TUGAS AKHIR - KI141502

PENGENALAN AKTIVITAS MANUSIA PADA VIDEO MENGGUNAKAN FITUR MATRIKS KOVARIAN

RINA WIJAYA KUSUMA WARDHANI
NRP 05111440000021

Dosen Pembimbing I
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

TUGAS AKHIR – KI141502

PENGENALAN AKTIVITAS MANUSIA PADA VIDEO MENGGUNAKAN FITUR MATRIKS KOVARIAN

RINA WIJAYA KUSUMA WARDHANI
NRP 05111440000021

Dosen Pembimbing I
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

Dosen Pembimbing II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

UNDERGRADUATE THESES – KI141502

HUMAN ACTIVITY RECOGNITION IN VIDEO USING FEATURE COVARIANCE MATRICES

RINA WIJAYA KUSUMA WARDHANI
NRP 05111440000021

Supervisor I
Dr. Eng. Nanik Suciati, S.Kom., M.Kom.

Supervisor II
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENGENALAN AKTIVITAS MANUSIA PADA VIDEO MENGGUNAKAN FITUR MATRIKS KOVARIAN

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh
RINA WIJAYA KUSUMA WARDHANI
NRP: 05111440000021

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr. Eng. Nanik Suciati, S.Kom., M.Kom.
   NIP: 197104281994122001 (Pembimbing 1)

2. Dini Adni Navastara, S.Kom., M.Sc.
   NIP: 198510172015042001 (Pembimbing 2)

SURABAYA
JUNI, 2018

[Halaman ini sengaja dikosongkan]

PENGENALAN AKTIVITAS MANUSIA PADA VIDEO MENGGUNAKAN FITUR MATRIKS KOVARIAN

Nama Mahasiswa	**: RINA WIJAYA KUSUMA WARDHANI**
NRP	**: 05111440000021**
Jurusan	**: Informatika FTIK-ITS**
Dosen Pembimbing 1	**: Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**
Dosen Pembimbing 2	**: Dini Adni Navastara, S.Kom., M.Sc.**

Abstrak

Saat ini, pengenalan aktivitas manusia pada data video merupakan persoalan yang cukup menantang di bidang visi komputer. Beberapa alat yang dapat digunakan untuk membantu pengenalan aktivitas manusia antara lain sensor *Accelerometer, Gyroscope, Camera,* dan GPS. Beberapa pengaplikasian pengenalan aktivitas manusia digunakan dalam sistem keamanan atau sarana hiburan.

Pada tugas akhir ini akan dilakukan pengembangan sistem pengenalan aktivitas manusia pada video yang berasal dari *Closed-Circuit Television* (CCTV) di Departemen Informatika ITS Surabaya. Sistem ini menggunakan fitur matriks kovarian dari fitur *optical flow* dimana fitur lokal ini dapat menangkap gerakan dinamis sebagai karakteristik dari aktivitas manusia pada video. Pengunaan fitur kovarian dilakukan untuk mengurangi jumlah *bag of feature vector* hasil dari ekstraksi fitur *optical flow* menjadi jauh lebih kecil namun tetap dapat merepresentasikan aktivitas manusia. Klasifikasi yang digunakan dalam sistem ini adalah *Nearest Neigbour Classifier*. Secara umum, pengembangan sistem ini dilakukan dalam lima tahapan proses yakni *preprocessing*, pengumpulan

fitur *optical flow*, perhitungan matriks kovarian, perhitungan matriks log kovarian, dan yang terakhir klasifikasi jenis aktivitas.

Uji coba menggunakan dataset CCTV menunjukkan bahwa metode yang digunakan pada tugas akhir ini memberikan hasil terbaik pada panjang *frame* L = 15, *overlapping frame* P = 1, dan pengambilan data pada waktu malam hari dengan nilai *accuracy* sebesar 95.86%, rata-rata *recall* sebesar 96.29%, dan rata-rata *precision* sebesar 96.01%, serta masing-masing hasil *recall* dan *precision* pada tiap kelas aktivitas sebesar 93.57% dan 94.60% untuk aktivitas berlari, 96.39% dan 96.24% untuk aktivitas berjalan, serta 98.91% dan 97.20% untuk aktivitas melambaikan kedua tangan.

Kata kunci: Pengenalan Aktivitas Manusia, Video, CCTV, Optical Flow, Matriks Kovarian.

HUMAN ACTIVITY RECOGNITION IN VIDEO USING FEATURE COVARIANCE MATRICES

Student's Name	**: RINA WIJAYA KUSUMA WARDHANI**
Student's ID	**: 05111440000021**
Department	**: Informatika FTIK-ITS**
First Advisor	**: Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**
Second Advisor	**: Dini Adni Navastara, S.Kom., M.Sc.**

Abstract

Currently, human activity recognition in video data are quite challenging in the field of computer vision. Some tools that can be used to help human activity recognition include Accelerometer, Gyroscope, Camera, and GPS. Many applications of human activity recognition in media systems or entertainment facilities.

In this final project, the development of human activity recognition system will be implemented on video from Closed-Circuit Television (CCTV) in Informatics Department ITS Surabaya. This system uses the covariance matrix feature of the optical flow feature which is the local features that can be used in the human video process. The use of covariance matrix features is done to reduce the number of bags of feature vectors so it becomes much smaller but still can represent human activity. The classification used within this system is the Nearest Neigbour Classification. In general, the development of this system is done in the process of pre-processing, optical flow flow, matrix calculation, covarian log matrix calculation, and activity classification.

Trials using CCTV dataset shows that the method used gives the best result in the use of frame Length L = 15, overlapping frame P = 1, and data collection time on night with an accuracy of 95.86%, average recall of 96.29%, and the average precision of 96.01%, with each recall and precision result in each activity class were 93.57% and 94.60% for running activity, 96.39% and 96.24% for walking activity, 98.91% and 97.20% for waving both hands activity.

Keywords: Human Activity Recognition, Video, CCTV, Optical Flow, Covarian Matrix.

KATA PENGANTAR

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang atas izin dan karunianya penulis dapat menyelesaikan Tugas Akhir yang berjudul ***"Pengenalan Aktivitas Manusia pada Video Menggunakan Fitur Matriks Kovarian".***

Dalam pengerjaan tugas akhir ini, penulis mendapatkan banyak sekali ilmu baru dan memperdalam ilmu-ilmu yang sebelumnya telah diajarkan selama masa perkuliahan di Teknik Informatika ITS.

Terselesaikannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Orang tua penulis Bapak Hadi Subito dan Ibu Trining Hastuti yang telah memberikan dukungan moral, spiritual dan material serta senantiasa memberikan doa demi kelancaran dan kemudahan penulis dalam mengerjakan tugas akhir.
3. Dr. Eng. Nanik Suciati, S.Kom., M.Kom. dan Dini Adni Navastara, S.Kom., M.Sc. selaku dosen pembimbing penulis yang telah memberi ide, nasihat dan arahan sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
4. Keempat saudara kandung penulis Mbak Rani, Mbak Nindy, Rini, dan Hito serta seluruh keluarga besar yang telah memberikan dukungan besar baik secara langsung maupun secara implisit.
5. Ilham Gurat Adillion, S.Kom. selaku pembimbing ketiga yang telah memberikan banyak pencerahan juga nasihat

kepada penulis serta menjadikan penulis bahan uji coba pelatihan menjadi dosen pembimbing sesungguhnya.

6. Rekan se-topik penelitian, Dzaky Zakiyal Fawwaz dan Fintanto Cendikia yang bersedia memberikan waktu untuk berdiskusi dalam pengerjaan tugas akhir.
7. Bayu Sektiaji dan Hari Setiawan yang bersedia meluangkan waktu untuk membantu dalam pembuatan dataset.
8. Ali Davito Hady dan Rafiar Rahmansyah yang telah menyemangati dan menghibur selama pengerjaan tugas akhir.
9. Teman-teman admin Laboratorium Pemrograman yang telah menemani dan memberikan semangat kepada penulis dalam menyelesaikan tugas akhir ini di Laboratorium Pemrograman.
10. Teman-teman LayToLele seperti Maya, Vina, Vennia, Sukma, Fitri, dan Dewayu
11. Serta semua pihak yang yang telah turut membantu penulis dalam menyelesaikan tugas akhir ini.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Juni 2018

Penulis

DAFTAR ISI

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

DAFTAR TABEL

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1 Latar Belakang

Saat ini, pengenalan aktivitas manusia pada data video merupakan persoalan yang cukup menantang di bidang visi komputer. Hal ini disebabkan oleh kompleksitas pada data video berupa halangan, kekacauan, interaksi objek ganda, perubahan cahaya, dan lain-lain. Selain itu, adanya persoalan pada proses pengambilan video seperti penyimpangan, pergerakan, dan sudut pandang kamera, serta kompleksitas pada aktivitas manusia itu sendiri sebagai objek non-rigid yang memiliki beragam kelas aktivitas [1]. Pengenalan aktivitas manusia pada data video telah diterapkan di beberapa bidang serta digunakan sebagai sistem keamanan atau sarana hiburan. Untuk itulah penulis ingin menerapkan pengenalan aktivitas manusia pada data video yang berasal dari *Closed-Circuit Television* (CCTV) di Departemen Informatika ITS Surabaya.

Ada dua komponen dasar pada setiap algoritma pengenalan aktivitas yang memengaruhi hasil akurasi dan efisiensi yaitu representasi model aktivitas dan metode klasifikasi yang digunakan. Model yang digunakan untuk merepresentasikan aktivitas dapat dibagi menjadi lima kategori yaitu *shape models, motion models*, *geometric human body models, interest-point models, dan dynamic models*. Metode klasifikasi standar yang paling sering digunakan antara lain *nearest-neighbor* (NN) classifier, *support vector machine* (SVM), dan *boosting* [1].

Pengenalan aktivitas dengan *shape-based models* menggunakan estimasi akurasi dari siluet objek yang bergerak di setiap *frame* video sehingga membentuk *silhouette tunnel*. Model ini bersifat invarian terhadap pencahayaan, warna, dan tekstur dari objek yang bergerak, namun tidak semua data video memiliki model ini [5]. Pengenalan aktivitas dengan *geometric human body models* menggunakan bagian tubuh manusia sebagai modelnya

sehingga akan sulit didapat dari data video dengan objek yang kecil dan *background* yang luas [8]. Pengenalan aktivitas dengan *interestpoint models* dilakukan dengan mengekstrak *intereset-point* dari pertimbangan informasi struktural dan mendeteksi *cuboids* di daerah yang memiliki kemungkinan besar mengalami pergerakan [9]. *Dynamic models* merupakan model yang paling baru digunakan dalam pengenalan aktivitas [10]. Model ini mendeskripsikan postur statis dari suatu aksi sebagai sebuah *state* dan mendeskripsikan *dynamics* (variasi temporal) dari suatu aksi menggunakan transisi model *state-space*. Model yang terakhir yaitu *motion models*. Model ini mengekstrak berbagai karakteristik pergerakan dan deformasi objek. Model ini menggunakan fitur kinematic yang berasal dari *optical flow* untuk merepresentasikan aktivitas [11-13]. *Motion-model* dikatakan sebagai atribut yang paling diskriminatif dalam merepresentasikan suatu aksi [1].

Tugas akhir ini mengusulkan penggunaan fitur matriks kovarian sebagai representasi ringkas dari kumpulan fitur lokal yang berasal dari *optical flow* sehingga memiliki dimensi yang lebih sedikit. Selain itu, tugas akhir ini menggunakan metode *nearest-neighbor* (NN) *classifier* untuk klasifikasi yang diketahui tahan terhadap keanekaragaman bentuk aktivitas, perubahan sudut pandang kamera, dan objek dengan resolusi rendah. Secara konseptual, *framework* ini cukup sederhana, membutuhkan ruang penyimpanan yang kecil, serta memiliki perhitungan yang memungkinkan untuk implementasi secara real time [1].

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara mendapatkan fitur lokal berbasis *optical flow* dari data video?
2. Bagaimana cara mendapatkan representasi model aktivitas manusia menggunakan fitur matriks kovarian?

3. Bagaimana cara klasifikasi aktivitas manusia menggunakan *nearest-neighbor* (NN) *classifier*?
4. Bagaimana mengetahui kinerja dari sistem pengenalan aktivitas manusia pada data video?

1.3 Batasan Masalah

Permasalan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

1. Data yang digunakan pada tugas akhir ini yaitu data video yang diperoleh dari CCTV di Departemen Informatika ITS Surabaya.
2. Aktivitas hanya dilakukan oleh manusia dan berobjek tunggal.
3. Aktivitas yang dapat dikenali berjumlah tiga yaitu aktivitas berlari, berjalan, dan melambaikan kedua tangan.
4. *Deployment* sistem dilakukan dalam bentuk *console*.

1.4 Tujuan

Tujuan dari tugas akhir ini adalah membuat sistem pengenalan aktivitas manusia pada video yang terekam oleh CCTV menggunakan fitur matriks kovarian.

1.5 Manfaat

Manfaat tugas akhir ini adalah sebagai langkah awal dalam membangun sistem pengawasan dan keamanan di Departemen Informatika ITS Surabaya melalui data rekaman video CCTV.

1.6 Metodologi

1.6.1 Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini akan mendeskripsikan dan membahas mengenai rencana pembuatan sistem pengenalan aktivitas manusia pada data video CCTV. Secara detil, proposal tugas akhir ini berisi tentang beberapa bagian yaitu latar belakang diajukannya tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu, dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir dan ringkasan isi yang membahas metode yang akan digunakan dalam tugas akhir. Sub bab metodologi merupakan penjelasan mengenai tahapan penyusunan tugas akhir. Terdapat pula sub bab jadwal pengerjaan yang menjelaskan jadwal pengerjaan tugas akhir dan di akhir bagian terdapat daftar pustaka untuk mencantumkan referensi yang digunakan dalam tugas akhir.

1.6.2 Studi Literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur sejumlah referensi tentang metode-metode yang akan digunakan dalam pengenalan aktivitas manusia pada data video. Informasi dan studi literatur tersebut didapat dari buku, internet, dan materi-materi kuliah yang berhubungan dengan metode yang digunakan.

1.6.3 Analisis dan Desain Perangkat Lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap

ini. Pada tahapan ini dilakukan desain sistem dan desain proses-proses yang ada.

1.6.4 Implementasi Perangkat Lunak

Pengembangan sistem dalam tugas akhir nantinya akan menggunakan matlab sebagai compiler dan library yang telah tersedia di matlab.

1.6.5 Pengujian dan Evaluasi

Pada tahapan ini dilakukan uji coba terhadap sistem yang telah dibuat. Pengujian ini betujuan untuk mengetahui unjuk kerja dari sistem pengenalan aktivitas pada data video. Pada uji coba ini, langkah pertama adalah melatih sistem dengan data training untuk membentuk model aktivitas. Langkah selanjutnya akan diberikan data uji untuk menguji model aktivitas yang telah terbentuk serta menghasilkan prediksi jenis aktivitas dari data uji. Parameter yang akan digunakan sebagai bahan uji coba adalah ukuran dari overlapping *frame* yang digunakan pada proses preprocessing video.

Setelah uji coba, dilakukan juga evaluasi terhadap kinerja dari sistem ini dengan cara melakukan pengukuran accuracy dan f-measure (precision dan recall) dari hasil uji coba yang berupa prediksi aktivitas terhadap aktivitas sebenarnya. Perhitungan accuracy dan f-measure dapat dilihat pada Tabel 1 dan persamaan (1.1), (1.2), dan (1.3). Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan untuk mengetahui kinerja sistem, mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

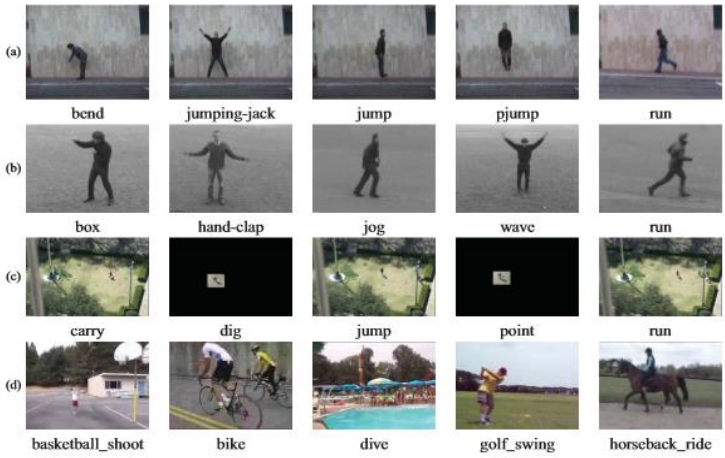
Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II
DASAR TEORI

2.1 Pengenalan Aktivitas Manusia

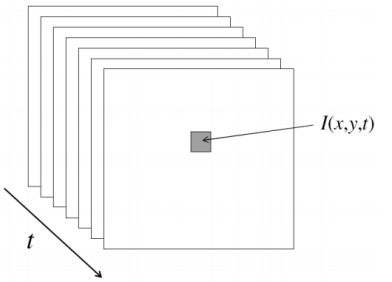
Pengenalan aktivitas manusia dapat dikelompokkan kedalam beberapa jenis. Pertama, pengenalan pada aktivitas sederhana yang dilakukan manusia seperti berjalan, naik tangga, turun tangga, jogging, dan lain-lain. Kedua, pada aktivitas kompleks yang biasanya mengkombinasikan aktivitas menjadi aktivitas dengan waktu yang lama, seperti; menunggu bus, mengemudi, dan lain-lain. Sebuah aktivitas juga dapat dilakukan hanya dengan beberapa anggota tubuh seperti; mengetik, melambaikan tangan, dan lain-lain. Aktivitas manusia juga dapat dikenali dengan menggunakan sensor seperti *Accelerometer, Gyroscope, Camera*, dan GPS [2]. Contoh aktivitas manusia pada data video ditunjukkan pada Gambar 2.1.



Gambar 2.1 Contoh Data Citra Aktivitas Manusia pada Video (a) Weizmann. (b) KTH. (c) UT_tower. (d) YouTube. [1]

2.2 Optical Flow

Optical flow merupakan fitur lokal yang dapat menangkap gerakan dinamis yang merupakan karakteristik aktivitas manusia dari data video dan terdiri dari 12 dimensi fitur. Pada beberapa tahun belakangan ini, sudah banyak penelitian yang membahas mengenai perhitungan *optical flow*. Pada kasus ini, yang akan digunakan adalah varian dari metode *Horn and Schunck*, yang mengoptimalkan fungsional berdasarkan residu dari batasan intensitas dan persyaratan regularisasi kehalusan [3].



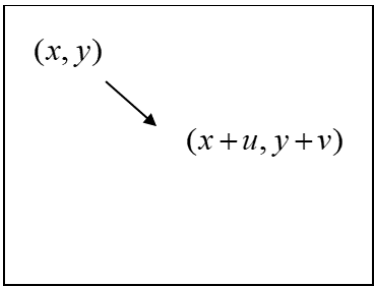
Gambar 2.2 Posisi Intensitas Terhadap Posisi Piksel dan *Frame*

$I(x, y, t)$ menunjukkan intensitas pencahayaan pada rangkaian video di posisi piksel (x, y, t) seperti ilustrasi pada Gambar 2. $\boldsymbol{u}(x, y, t)$ mewakili vector optical flow yang sesuai, dimana $\boldsymbol{u} = (u, v)^T$**.** Berdasarkan $I(x, y, t)$ dan $\boldsymbol{u}(x, y, t)$, dapat didefinisikan vector fitur lokal $\boldsymbol{f}(x, y, t)$ pada persamaan (2.1).

$$\boldsymbol{f}(x, y, t) := [x, y, t, I_t, u, v, u_t, v_t, Div, Vor, Gten, Sten]^T \quad (2.1)$$

dimana $(x, y, t)^T \in A$ (himpunan semua koordinat piksel pada segmen video), I_t adalah turunan parsial pertama dari $I(x, y, t)$ terhadap *frame* t, sehingga I_t dapat didefinisakan seperti pada persamaan (2.2).

$$I_t = \frac{\partial I(x,y,t)}{\partial t} \tag{2.2}$$



Gambar 2.3 Ilustrasi Perpindahan Intensitas pada *Frame*

u dan v adalah komponen *optical flow* yang merupakan perpindahan posisi piksel (x, y) dengan intensitas I berdasarkan t seperti ilustrasi pada Gambar 3. Diasumsikan intensitas piksel antara *frame* satu dengan *frame L*ainnya bersifat konstan, sehingga dapat didefinisikan seperti pada persamaan (2.3). Metode perhitungan fitur u dan v menggunakan metode *Horn and Schunk* didefinisikan berurutan pada persamaan (2.4) dan (2.5). Sementara u_t dan v_t adalah turunan parsial pertama dari u dan v terhadap t sehingga u_t dan v_t dapat didefinisikan seperti pada persamaan (2.6) dan (2.7)

$$I(x, y, t-1) = I(x+u, y+v, t) \tag{2.3}$$

$$u^{k+1} = \bar{u}^k - \frac{I_x\,(I_x\bar{u}^k + I_y\bar{v}^k + I_t\,)}{\alpha^2 + {I_x}^2 + {I_y}^2} \tag{2.4}$$

$$v^{k+1} = \bar{v}^k - \frac{I_y\,(I_x\bar{u}^k + I_y\bar{v}^k + I_t\,)}{\alpha^2 + {I_x}^2 + {I_y}^2} \tag{2.5}$$

$$u_t = \frac{\partial u(x,y,t)}{\partial t} \tag{2.6}$$

$$v_t = \frac{\partial v(x,y,t)}{\partial t} \tag{2.7}$$

Div (*divergence*) adalah perbedaan spasial dari *flow field* yang ada pada setiap posisi piksel. *Divergence* menangkap jumlah ekspansi lokal yang dapat mengindikasikan perbedaan aktivitas. *Divergence* didefinisikan pada persamaan (2.8) sebagai berikut.

$$Div(x,y,t) = \frac{\partial u(x,y,t)}{\partial x} + \frac{\partial v(x,y,t)}{\partial y} \tag{2.8}$$

Vor (*vorticity*) digunakan untuk mengukur putaran lokal di sekitar sumbu yang tegak lurus terhadap bidang *flow field*. Pada konteks *optical flow*, *vorticity* berpotensi menangkap gerakan melingkar lokal dari objek yang bergerak. *Vorticity* didefinisikan pada persamaan (2.9) sebagai berikut.

$$Vor(x,y,t) = \frac{\partial v(x,y,t)}{\partial x} - \frac{\partial u(x,y,t)}{\partial y} \tag{2.9}$$

Untuk menggambarkan $Gten$ dan $Sten$ ada dua matriks yang berperan, yaitu tensor gradien dari *optical flow* $\nabla \boldsymbol{u}(x,y,t)$ dan laju regangan tensor $S(x,y,t)$ yang didefinisikan pada berurutan pada persamaan (2.10) dan (2.11) sebagai berikut.

$$\nabla \boldsymbol{u}(x,y,t) = \begin{matrix} \frac{\partial u(x,y,t)}{\partial x} & \frac{\partial v(x,y,t)}{\partial y} \\ \frac{\partial v(x,y,t)}{\partial x} & \frac{\partial u(x,y,t)}{\partial y} \end{matrix} \tag{2.10}$$

$$S(x,y,t) = \frac{1}{2}(\nabla \boldsymbol{u}(x,y,t) + \nabla^T \boldsymbol{u}(x,y,t)) \quad (2.11)$$

$Gten$ dan $Sten$ adalah invarian tensor yang tetap konstan tidak peduli sistem koordinat apa yang mereka rujuk. $Gten$ dan $Sten$ juga merupakan sifat skalar yang menggabungkan komponen tensor gradien sehingga menghasilkan struktur lokal. $Gten$ dan $Sten$ didefinisikan berurutan pada persamaan (2.12) dan (2.13) sebagai berikut.

$$Gten(x,y,t) = \frac{1}{2}(tr^2\big(\nabla \boldsymbol{u}(x,y,t)\big) - tr(\nabla^T \boldsymbol{u}(x,y,t))) \quad (2.12)$$

$$Sten(x,y,t) = \frac{1}{2}(tr^2\big(S(x,y,t)\big) - tr(S^2(x,y,t))) \quad (2.13)$$

dimana $tr(\cdot)$ menunjukkan *trace operation*.

2.3 Matriks Kovarian

Sampel video biasanya berdimensi sangat tinggi. Tentu sangat tidak praktis mempelajari struktur global sampel video *learning* dan membangun klasifikasi secara langsung di ruang berdimensi tinggi. Oleh karena itu, dilakukanlah pendekatan pemodelan "*bag of dense local feature vector*" dimana serangkaian fitur lokal diekstrak dari data video untuk merepresentasikan aktivitas. Namun, dimensi sekumpulan vector fitur lokal tersebut bahkan lebih besar daripada sampel video yang diekstraksi karena jumlah piksel yang dikali dengan ukuran vector fitur tersebut. Sehingga diperlukannya pengurangan dimensi.

Fitur matriks kovarian dapat menyediakan representasi yang sangat baik untuk pengenalan aktivitas. Selain sederhana dan efektif, matriks kovarian dari fitur lokal memiliki ruang penyimpanan dan kebutuhan proses yang kecil. [4][5]

$F = \{f_n\}$ menunjukkan "*bag of feature vector*" dan N adalah ukuran dari F. Matriks kovarian C dari F dapat didefinisikan seperti pada persamaan (2.14) sebagai berikut.

$$C \coloneqq \frac{1}{N}\sum_{n=1}^{N}(f_n - \mu)(f_n - \mu)^T \tag{2.14}$$

dimana μ adalah mean dari vector fitur yang didefinisikan pada persamaan (2.15) sebagai berikut.

$$\mu = \frac{1}{N}\sum_{n=1}^{N} f_n \tag{2.15}$$

Matriks kovarian menyediakan cara alami untuk menggabungkan beberapa vector fitur. Dimensi dari matriks kovarian hanya berhubungan dengan dimensi dari vector fitur. Jika f_n berdimensi d, maka C merupakan matriks berukuran $d \times d$. Karena d jauh lebih kecil dari N, matriks C akan memiliki dimensi yang jauh lebih kecil dibandingkan dengan dimensi dari "*bag of feature vector*" yang membutuhkan $N \times d$ dimensi.

2.4 Normalisasi

Normalisasi merupakan teknik untuk menstandarkan atau menyamakan rentang data sehingga tidak ada satu atribut yang terlalu dominan atas atribut yang lain. Salah satu teknik normalisasi adalah normalisasi skala.

Normalisasi skala merupakan teknik penstandaran data pada suatu rentang tertentu, umumnya 0-1. Diketahui nilai maksimal dari suatu piksel citra sebagai $Nmax$ dan nilai minimal dari suatu piksel citra adalah $Nmin$. Citra yang dinormalisasi disimbolkan dengan N. Nilai hasil normalisasi didapatkan dari rumus

perhitungan normalisasi skala ditunjukkan pada persamaan (2.16) sebagai berikut.

$$Normalisasi = \frac{N - Nmin}{Nmax - Nmin} \tag{2.16}$$

2.5 Matriks Log-Kovarian

Himpunan dari semua matriks kovarian dengan ukuran tertentu tidak membentuk sebuah ruang vector karena tidak diikuti oleh perkalian dengan scalar negatif, sehingga membentuk "*closed convex cone*" [6]. Kebanyakan dari algoritma *machine learning* bekerja dengan fitur yang berada pada ruang Euclidean, bukan *convex cone*. Oleh karena itu, diperlukannya pemetaan *convex cone* pada matriks kovarian ke dalam ruang vector dengan menggunakan logaritma dari matriks kovarian [7]. Perhitungan matriks kovarian dijelaskan pada persamaan (2.17) sebagai berikut.

$$C = VDV^T \tag{2.17}$$

dimana V adalah *orthonormal eigenvectors* dan D adalah matriks diagonal dari *eigenvalues*. Selanjutnya $\log(C)$ didefinisikan seperti pada persamaan (2.18) sebagai berikut.

$$\log(C) \coloneqq V\widetilde{D}V^T \tag{2.18}$$

Dimana $\widetilde{D}$ adalah matriks diagonal dari matriks D dengan mengganti entri diagonal matriks D dengan nilai logaritmanya.

2.6 *Nearest-Neighbor* (NN) *Classification*

Nearest-neighbor classification merupakan algoritma yang paling banyak digunakan untuk klasifikasi. Algoritma ini cukup sederhana dan mudah; diberikan sebuah sampel data testing.

Temukan sampel yang paling mirip dengan sampel data testing tersebut pada himpunan data training, dimana kemiripan diukur dari suatu perhitungan jarak antara keduanya. Selanjutnya, beri label kelas pada sampel data testing tersebut.

Keberhasilan dari klasifikasi menggunakan NN classifier bergantung pada metrik jarak yang digunakan. Pada kasus ini, metric yang akan digunakan adalah pengukuran jarak Euclidean antara logaritma dari representasi aktivitas matriks kovarian.

$$\rho(C_1, C_2) \coloneqq \| \log(C_1) - \log(C_2) \|_2 \qquad (2.19)$$

dimana $\log(\cdot)$ adalah logaritma matriks kovarian dan $\| \cdot \|_2$ menunjukkan norma *Frobenius* pada matriks. Jarak $\rho(C_1, C_2)$ yang didefinisikan pada persamaan (2.19) dapat ditunjukkan sebagai metrik *Riemannian* untuk jenis matriks kovarians. Metrik Ini disebut sebagai metrik *log-Euclidean*.

2.7 Confusion Matrix

Confusion matrix merupakan matriks yang mengandung informasi tentang kelas sebenarnya dan prediksi yang dihasilkan oleh sistem klasifikasi. *Confusion matrix* banyak digunakan untuk menguji performa dari suatu metode klasifikasi. Struktur *confusion matrix* untuk tiga kelas ditunjukkan pada Tabel 2.1 sebagai berikut.

Tabel 2.1 *Confusion Matrix* untuk 3 Kelas

		Nilai Prediksi		
		1	***2***	***3***
Nilai Sebenarnya	***1***	$TP1$	$F1.2$	$F1.3$
	2	$F2.1$	$TP2$	$F2.3$
	3	$F3.1$	$F3.2$	$TP3$

Keterangan:

TP = Jumlah data uji yang kelas prediksinya sama dengan kelas yang sebenarnya.
F = Jumlah kelas prediksi yang tidak sesuai dengan kelas sebenarnya.

Beberapa nilai evaluasi yang bisa dihitung berdasarkan confusion matrix untuk mengetahui performa *classifier* yaitu *accuracy*, *recall*, dan *precision*. *Accuracy* adalah perbandingan jumlah total data yang prediksi kelas hasil klasifikasinya sesuai dengan *ground truth* terhadap seluruh data. Rumus perhitungan dari akurasi ditunjukkan pada persamaan (2.20) sebagai berikut.

$$Accuracy = \frac{TP1+TP2+TP3}{TP1+TP2+TP3+F1.2+F1.3+F2.1+F2.3+F3.1+F3.2} \quad (2.20)$$

Recall adalah perbandingan dari jumlah data yang prediksi kelas hasil klasifikasinya sesuai dengan *ground truth* terhadap seluruh data berkelas benar pada ground truth. Rumus perhitungan *recall* ditunjukkan pada persamaan (2.21), (2.22), dan (2.23) sebagai berikut.

$$Recall1 = \frac{TP1}{TP1+F1.2+F1.3} \quad (2.21)$$

$$Recall2 = \frac{TP2}{TP2+F2.1+F2.3} \quad (2.22)$$

$$Recall3 = \frac{TP3}{TP3+F3.1+F3.2} \quad (2.23)$$

Precision adalah adalah perbandingan dari jumlah data yang prediksi kelas hasil klasifikasinya sesuai dengan *ground truth* terhadap keseluruhan data yang terklarifikasi benar. Rumus

perhitungan *precision* ditunjukkan pada persamaan (2.24), (2.25), dan (2.26) sebagai berikut.

$$Precision1 = \frac{TP1}{TP1+F2.1+F3.1} \quad (2.24)$$

$$Precision2 = \frac{TP2}{TP2+F1.2+F3.2} \quad (2.25)$$

$$Precision3 = \frac{TP3}{TP3+F1.3+F2.3} \quad (2.26)$$

BAB III
ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas perancangan sistem pengenalan aktivitas manusia pada video. Perancagan yang dibuat meliputi perancangan sistem, perancangan data, dan perancangan proses. Hasil dari proses ini berupa diagram yang akan digunakan sebagai acuan untuk proses implementasi sistem.

3.1 Lingkungan Desain dan Implementasi

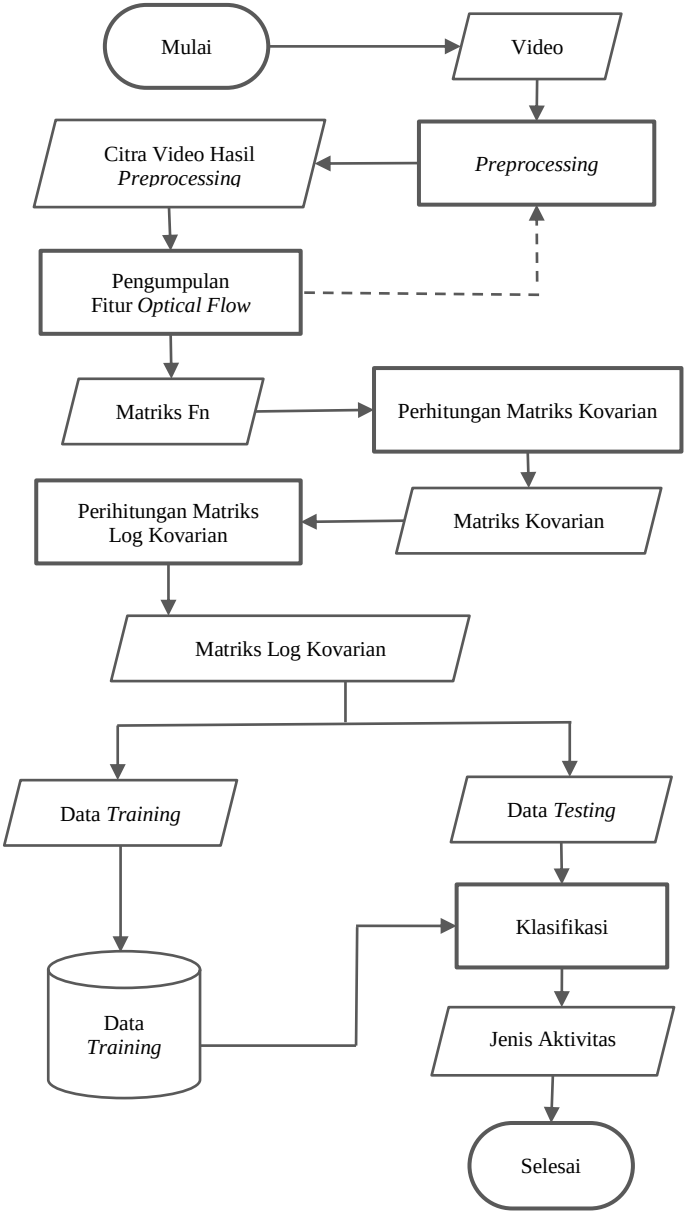
Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam desain dan implementasi pembuatan sistem disebutkan dalam Tabel 3.1.

Tabel 3.1 Lingkungan Perancangan Sistem

No.	Jenis	Spesifikasi
1	Prosesor	Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz 3.40 GHz
2	Memori	4.00 GB
3	Sistem Operasi	Windows 10 Enterprise 64-bit
4	Perangkat Lunak	Matlab R2018a Ms. Excel 2016 Ms.Word 2016

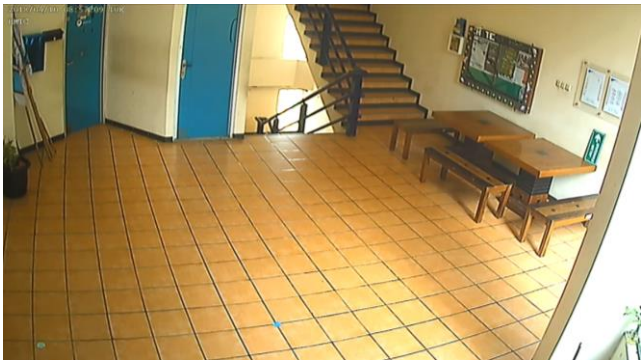
3.2 Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan proses sistem pengenalan aktivitas manusia pada video secara keseluruhan. Rancangan sistem ini ditunjukan dalam bentuk diagram alir pada Gambar 3.1 yang terdiri dari satu data masukan, lima proses, dan satu data keluaran.



Gambar 3.1 Diagram Alir Sistem secara Keseluruhan

Data masukan yang digunakan adalah video yang berasal dar *CCTV* di Departemen Informatika ITS lantai 3, sebagai salah satu tempat yang paling sering terjadi aktivitas manusia seperti berjalan atau berlari. Contoh potongan video *CCTV* tersebut ditunjukkan pada Gambar 3.2.

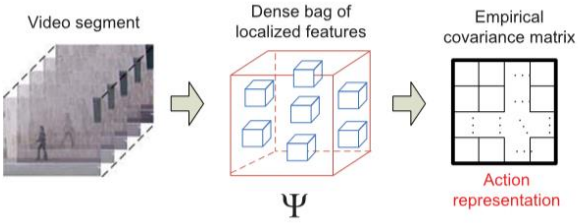


Gambar 3.2 View CCTV Departemen Informatika ITS

Proses *preprocessing* adalah proses untuk menyiapkan citra video sebelum masuk ke proses pengumpulan fitur. Persiapan tersebut berupa pengumpulan video berdasarkan aktivitas, *resize* video, dan mengubah video menjadi citra *greyscale*. Hasil *preprocessing* tersebut digunakan sebagai data masukan proses pengumpulan fitur.

Proses pengumpulan fitur terdiri dari proses perhitungan dan penggabungan fitur-fitur *optical flow*. Fitur *optical flow* terdiri dari 12 dimensi fitur yang diambil dari setiap piksel citra video. Sehinga hasil dari proses ini berupa *bag of localized features* berukuran N × 12 dimana N adalah jumlah seluruh piksel dalam satu segmen video. *Bag of localized features* inilah yang akan menjadi data masukan pada proses perhitungan matriks kovarian sebagai representasi aktivitas.

Seperti ilustrasi pada Gambar 3.3, setelah fitur-fitur *optical flow* dari suatu segmen video terkumpul, selanjutnya adalah merepresentasikannya menjadi sebuah aksi atau aktivitas dengan



Gambar 3.3 Ilustrasi Perubahan Dimensi Video

cara menghitung matriks kovarian dari fitur-fitur lokal tersebut. Hal ini bertujuan untuk mendapatkan dimensi yang jauh lebih kecil.

Setelah diperoleh fitur dalam bentuk matriks kovarian, dilanjutkan dengan menghitung logaritma dari matriks kovarian tersebut. Hal ini bertujuan untuk memetakan bentuk matriks yang sebelumnya "*convex cone*" menjadi bentuk ruang *vector* sehingga dapat digunakan oleh *machine learning* untuk proses klasifikasi selanjutnya.

Proses klasifikasi dilakukan menggunakan *Nearest-Neighbor classifier* dengan jarak Euclidean sebagai *distance metric*-nya.

3.3 Perancangan Data

Perancangan data dilakukan untuk memastikan pengoperasian sistem berjalan dengan benar. Data masukan (*input*) adalah data yang diperlukan dalam pengoperasian sistem dan data keluaran (*output*) adalah data yang dihasilkan oleh sistem serta akan digunakan oleh pengguna.

Data masukan pada sistem pengenalan aktivitas manusia adalah data video berukuran 636 × 360 piksel dengan citra warna RGB dan memiliki *framerate* sebesar 25 *fps*. Waktu yang digunakan untuk menggambarkan suatu gerakan adalah sekitar 0,4

— 0,8 detik yang berarti sepanjang 10 — 20 *frame* untuk setiap segmennya.

Gambar 3.4 Titik dan Arah Gerak Aktivitas

Satu data video menampilkan satu objek yang melakukan satu jenis aktivitas. Aktivitas yang dilakukan yaitu aktivitas berlari, berjalan, dan melambaikan kedua tangan. Masing-masing aktivitas tersebut terdiri dari beberapa ragam pola arah dan titik gerakan pada latar video. Seperti yang ditunjukkan pada Gambar 3.4, ada enam arah aktivitas berlari, enam arah aktivitas berjalan, dan empat titik aktivitas melambaikan kedua tangan. Masing-masing gerakan aktivitas dilakukan oleh enam orang.

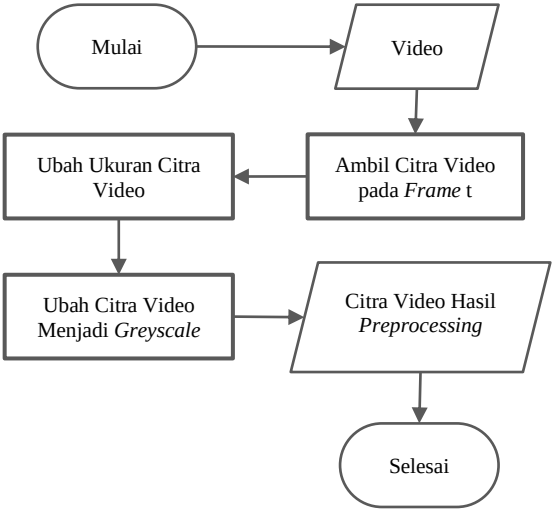
Data keluaran sistem pengenalan aktivitas manusia merupakan prediksi aktivitas yang dilakukan objek pada data masukan baik itu berlari, berjalan, atau melambaikan kedua tangan.

3.4 Perancangan Proses

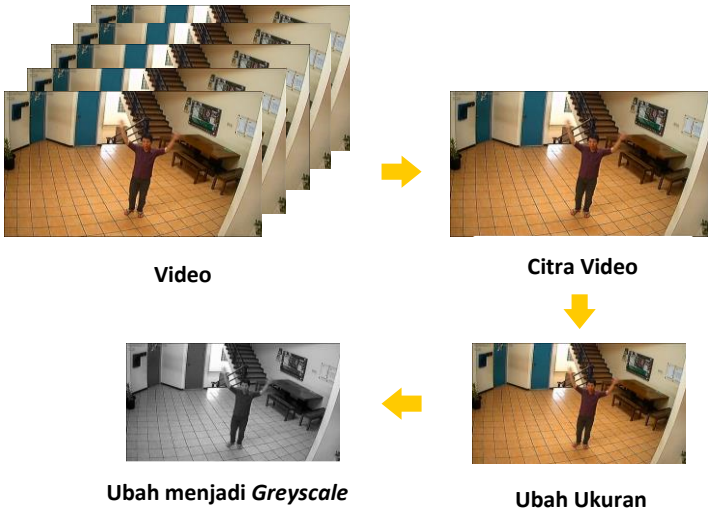
Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada sistem pengenalan aktivitas manusia. Bagian dari setiap proses utama sistem secara keseluruhan dapat dilihat pada Gambar 3.1.

3.4.1 *Preprocessing*

Proses pertama yang dilakukan adalah *preprocessing* yaitu mengambil citra video pada *frame* t, mengubah ukuran citra video yang sebelumnya 1080 × 720 piksel diperkecil menjadi 636 × 360 piksel, serta mengubah citra video dari RGB menjadi *greyscale*. Hal tersebut dilakukan untuk mengoptimalkan kerja sistem. Diagram alir proses ini ditunjukkan pada Gambar 3.5, serta ilustrasi proses ditunjukkan pada Gambar 3.6.



Gambar 3.5 Diagram Alir Proses *Preprocessing*



Gambar 3.6 Ilustrasi Proses *Preprocessing*

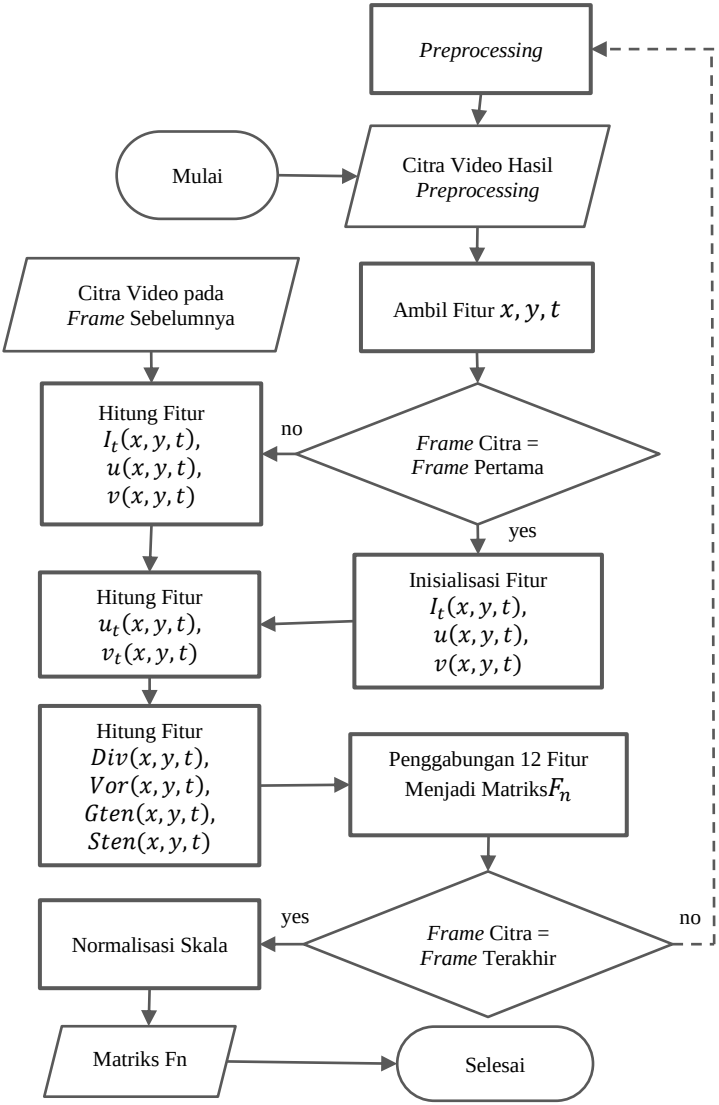
3.4.2 Pengumpulan Fitur *Optical Flow*

Proses pengumpulan fitur pada video hasil *preprocessing* terdiri dari proses perhitungan fitur *optical flow* pada citra video dan proses penggabungan fitur-fitur tersebut dalam satu segmen video menjadi sebuah matriks Fn. Fitur-fitur *optical flow* tersebut terdiri dari 12 dimensi fitur yang akan diambil dari setiap piksel pada citra video.

Fitur-fitur tersebut antara lain x, y, t, I_t, u, v, u_t, v_t, Div, Vor, $Gten$, dan $Sten$ seperti yang dijelaskan pada Tabel 3.2. Fitur-fitur tersebut akan dikumpulkan menjadi satu matriks Fn pada setiap segmen video dengan panjang L *frame*, sehingga satu segmen video akan memiliki ukuran dimensi fitur sebesar $12 \times 636 \times 360 \times L$. Diagram alir proses ini ditunjukkan pada Gambar 3.7.

Tabel 3.2 Penjelasan Ringkas Fitur *Optical Flow*

No.	Fitur	Deskripsi	Persamaan
1	x	Posisi piksel x	-
2	y	Posisi piksel y	-
3	t	Posisi *frame*	-
4	$u(x, y, t)$	Perpindahan posisi piksel x dengan intensitas $I(x, y, t)$ terhadap t	(2.4)
5	$v(x, y, t)$	Perpindahan posisi piksel y dengan intensitas $I(x, y, t)$ terhadap t	(2.5)
6	$I_t(x, y, t)$	Turunan parsial pertama dari intensitas $I(x, y, t)$ terhadap t	(2.2)
7	$u_t(x, y, t)$	Turunan parsial pertama dari u terhadap t	(2.6)
8	$v_t(x, y, t)$	Turunan parsial pertama dari v terhadap t	(2.7)
9	$Div(x, y, t)$ (*Divergent*)	Perbedaan spasial dari *flow field* yang ada pada setiap posisi piksel	(2.8)
10	$Vor(x, y, t)$ (*Vorticity*)	Putaran / gerakan melingkar lokal di sekitar sumbu yang tegak lurus terhadap bidang *flow field*	(2.9)
11	$Gten(x, y, t)$	*Gradient tensor* yang tetap konstan tidak peduli sistem koordinat apa yang digunakan (invarian)	(2.12)
12	$Sten(x, y, t)$	*Strain tensor* yang tetap konstan tidak peduli sistem koordinat apa yang digunakan (invarian)	(2.13)



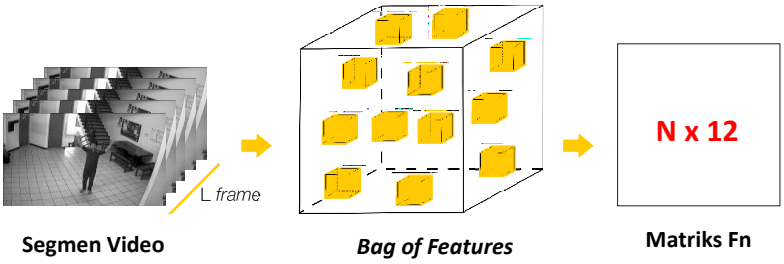
Gambar 3.7 Diagram Alir Proses Pengumpulan Fitur *Optical Flow*

Proses pertama yaitu mengambil fitur x, y, dan t yang menunjukkan posisi-posisi piksel dan *frame* pada citra video dengan cara membuat tiga matriks berukuran citra video yang masing-masing matriksnya berisi informasi posisi piksel x, posisi piksel y, dan posisi $frame\ t$ dari setiap piksel citra video tersebut. Selanjutnya adalah pengambilan fitur u, v, dan I_t dengan terlebih dahulu mengecek apakah citra video merupakan citra pertama dalam satu segmen atau bukan. Jika citra masukkan termasuk citra pertama, maka akan dilakukan inisialisasi fitur u, v, dan I_t. Jika bukan, maka akan dilakukan perhitungan fitur tersebut menggunakan data fitur u dan v dari citra video dari *frame* sebelumnya.

Proses selanjutnya adalah perhitungan fitur u_t dan v_t yang merupakan turunan parsial masing-masing fitur u dan v terhadap posisi *frame* t. Dilanjutkan dengan perhitungan fitur Div, Vor, $Gten$, dan $Sten$ menggunakan data fitur u_t dan v_t.

Selanjutnya, ke-12 fitur masing-masing piksel pada citra video digabungkan menjadi sebuah matriks Fn. Setelah itu dilakukan pengecekan apakah citra video tersebut merupakan citra terakhir dalam satu segmen atau bukan. Jika citra masukan bukan termasuk citra terakhir, maka dilakukan proses pengulangan ke proses *preprocessing* yaitu pengambilan citra video pada *frame* selanjutnya.

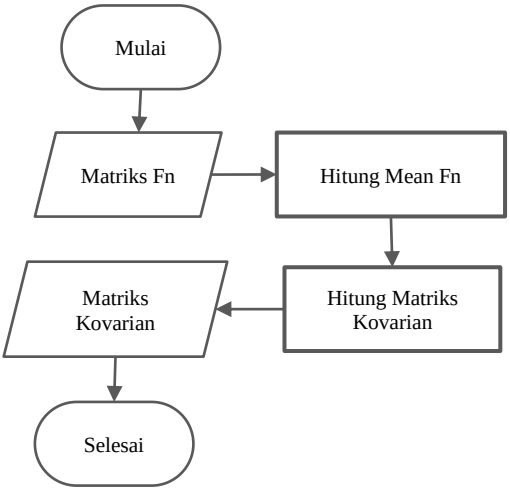
Setelah terkumpul fitur-fitur *optical flow* pada citra-citra video dalam satu segmen sepanjang *L frame* dalam bentuk matriks Fn, maka dilanjutkan ke proses terakhir yaitu normalisasi matriks Fn menggunakan normalisasi skala. Ilustrasi proses pengumpulan fitur *optical flow* ditunjukkan pada Gambar 3.8.



Gambar 3.8 Ilustrasi Proses Pengumpulan Fitur Optical Flow

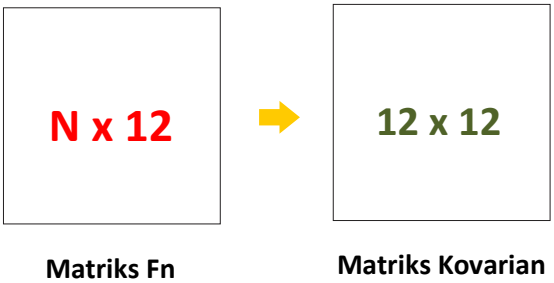
3.4.3 Perhitungan Matriks Kovarian

Perhitungan Matriks Kovarian adalah proses pengolahan matriks Fn menjadi suatu data yang dapat digunakan untuk merepresentasikan sebuah aktivitas menggunakan matriks kovariannya. Hal tersebut dilakukan untuk mengurangi ukuran dimensi matriks Fn menjadi jauh lebih kecil yaitu dari ukuran $12 \times 636 \times 360 \times L$ menjadi 12×12. Diagram alir proses ini ditunjukkan pada Gambar 3.9.



Gambar 3.9 Diagram Alir Proses Perhitungan Matriks Kovarian

Proses pertama adalah dengan menghitung mean matriks Fn terhadap ke-12 fitur optical flow, dilanjutkan dengan menghitung mariks kovariannya. Detail perhitungan proses ini ditunjukkan pada persamaan (2.14) dan (2.15). Ilustrasi proses perhitungan matriks kovarian ditunjukkan pada Gambar 3.10.

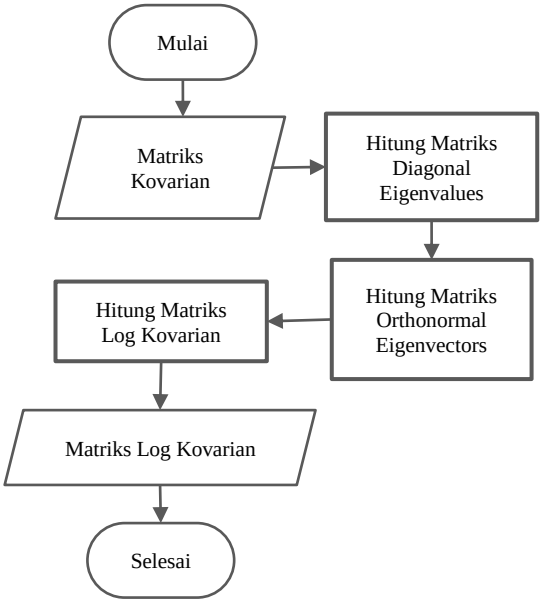


Gambar 3.10 Ilustrasi Proses Perhitungan Matriks Kovarian

3.4.4 Perhitungan Matriks Log Kovarian

Proses selanjutnya adalah menghitung logaritma dari matriks kovarian. Hal ini dilakukan untuk membentuk matriks menjadi suatu ruang vector sehingga dapat digunakan oleh proses selanjutnya yakni klasifikasi. Diagram alir proses ini ditunjukkan pada Gambar 3.11.

Proses pertama adalah perhitungan matriks diagonal eigenvalues dari matriks kovarian. Dilanjutkan dengan perhitungan matriks orthonormal eigenvectors dari matriks kovarian. Lalu perhitungan matriks log kovarian menggunakan perkalian matriks antara matriks diagonal eigenvalues dan matriks orthonormal eigenvectors tersebut. Detail perhitungan proses ini ditunjukkan pada persamaan (2.17) dan (2.18). Hasil dari proses inilah yang menjadi data untuk proses pembuatan model klasifikasi dan proses klasifikasi selanjutnya.

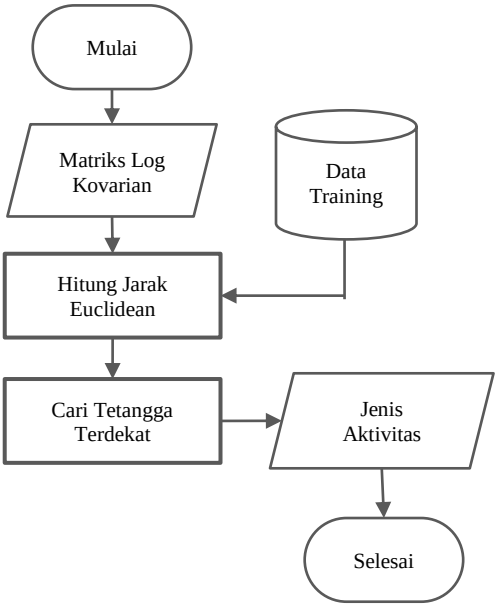


Gambar 3.11 Diagram Alir Proses Perhitungan Matriks Log Kovarian

3.4.5 Klasifikasi Akivitas

Proses terakhir adalah proses klasifikasi untuk mengetahui jenis aktivitas yang dilakukan menggunakan *Nearest-Neighbor classifier*. Data masukan pada proses ini yaitu matriks log kovarian yang merupakan data keluaran proses sebelumnya sebagai data testing, serta data training yang telah tersimpan sebelumnya dalam sistem. Diagram alir proses ini ditunjukkan pada Gambar 3.12.

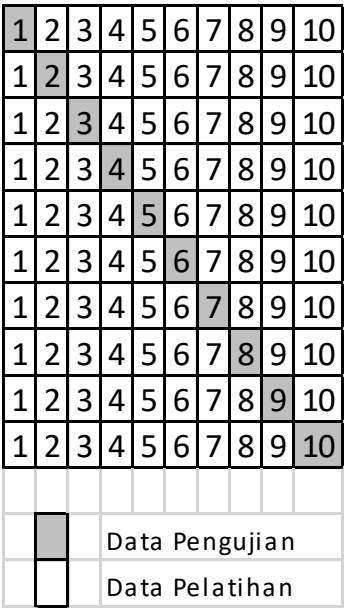
Proses pertama adalah perhitungan jarak antara matriks log kovarian data training menggunakan jarak euclidean sebagai *distance metric*-nya atau dalam hal ini disebut sebagai *log-Euclidean metric*. Setelah itu, nilai jarak setiap log kovarian diurutkan dan diambil yang paling kecil untuk mendapatkan jarak terpendek. Jarak terpendek tersebut yang akan menunjukkan jenis aktivitas apa yang paling mirip sebagai hasil klasifikasi.



Gambar 3.12 Diagram Alir Proses Klasifikasi

3.4.6 Pengukuran Kinerja Sistem

Proses pengukuran kerja sistem dilakukan menggunakan metode 10-*fold cross validation* karena metode ini dapat mengurangi waktu komputasi dengan tetap menjaga keakuratan estimasi. Dalam 10-*fold cross validation*, data dibagi menjadi 10-*fold* berukuran kira-kira sama, sehingga kita memiliki 10 subset data untuk mengevaluasi kinerja model atau algoritma. Untuk masing-masing dari 10 subset data tersebut, metode ini akan menggunakan 9-*fold* untuk pelatihan dan 1-*fold* untuk pengujian seperti diilustrasikan pada Gambar 3.13.



Gambar 3.13 Skema 10-*fold cross validation*

[Halaman ini sengaja dikosongkan]

BAB IV
IMPLEMENTASI

Pembahasan implementasi pada bab ini meliputi deskripsi lingkungan tahap implementasi, proses-proses pada tahap implementasi yang dikerjakan, beserta penjelasan fungsi-fungsinya dalam bentuk kode sumber.

4.1 Lingkungan Implementasi

Beberapa perangkat pendukung yang digunakan pada proses implementasi sistem pengenalan aktivitas ialah sebagai berikut.

4.1.1 Perangkat Keras

Lingkungan implementasi pada tugas akhir ini adalah sebuah *personal computer* (PC). Spesifikasi dari PC yang digunakan pada tugas akhir ini adalah memiliki prosesor Intel Core i3-3240 dengan kecepatan 3,40 GHz dan Random Access Memory (RAM) untuk proses menjalankan program sebesar 4,00 GB.

4.1.2 Perangkat Lunak

Lingkungan implementasi pada tugas akhir ini adalah sebuah personal computer (PC). Spesifikasi PC dari sisi perangkat lunak menggunakan software MATLAB R2018a, Microsoft Office Excel dan Microsoft Office Word 2016.

4.2 Implementasi Preprocessing

Proses *preprocessing* pada tugas akhir ini terdiri dari tiga tahap yaitu mengambil citra video pada *frame* t, mengubah ukuran citra, dan mengubahnya menjadi citra *greyscale*. Implementasi proses *preprocessing* ditunjukkan pada Kode Sumber 4.1.

```
1  vid = VideoReader(videoFileName);
2  im = read(vid,t);
3  imResize = imresize(im,scale);
4  imNew = single(rgb2gray(imResize));
```

Kode Sumber 4.1 Implementasi *Preprocessing*

Baris 1 menunjukkan pembacaan file video. Baris 2 menunjukkan pengambilan citra video pada *frame* ke-t dari file video. Dilanjutkan dengan perubahan ukuran citra video pada baris 3 dan perubahan citra video menjadi *greyscale* pada baris 4. Proses ini dilakukan untuk mengoptimalkan kerja sistem.

4.3 Implementasi Pengumpulan Fitur *Optical Flow*

Proses pengumpulan fitur *optical flow* pada tugas akhir ini terdiri dari beberapa tahap di antaranya perhitungan fitur dan penggabungan fitur dalam satu segmen. Tahap perhitungan fitur adalah tahap pengumpulan 12 fitur *optical flow* pada data citra video, yang selanjutnya akan digabung dengan fitur-fitur pada citra video lainnya dalam segmen yang sama menjadi matriks Fn. Penjelasan lebih rinci dari masing-masing tahapan tersebut adalah sebagai berikut.

4.3.1 Pengambilan Fitur x, y, dan t

Pengambilan fitur x, y, dan t yang menunjukkan posisi-posisi piksel dan *frame* pada citra video. Implementasi proses ini ditunjukkan pada Kode Sumber 4.2. Oleh karena pengumpulan fitur *optical flow* dilakukan pada setiap piksel citra video, maka dibuatlah masing-masing sebuah matriks yang berisi posisi piksel x, posisi piksel y, serta posisi *frame* t seukuran dengan citra video.

Baris 1 menunjukkan pengambilan ukuran citra video. Baris 2 dan 3 menunjukkan pembuatan matriks fitur x, baris 4 dan

5 menunjukkan pembuatan matriks fitur y, dan baris 6 menunjukkan pembuatan matriks fitur t.

```
1  [m,n] = size(imNew);
2  row = reshape(1:m,m,[]);
3  matX = row*ones(1,n);
4  col = reshape(1:n,[],n);
5  matY = ones(m,1)*col;
6  matT = ones(m,n)*t;
```

Kode Sumber 4.2 Implementasi Pengambilan Fitur x, y, dan t

4.3.2 Inisialisasi

Sebelum melakukan pengambilan fitur u, v, dan I_t, terlebih dahulu dicek apakah citra video merupakan citra pertama dalam suatu segmen atau bukan. Jika citra masukkan termasuk citra pertama, maka akan dilakukan inisialisasi fitur-fitur yang dibutuhkan dalam perhitungan fitur *optical flow* selanjutnya. Implementasi proses inisialisasi ditunjukkan pada Kode Sumber 4.3.

```
1  It = zeros(size(imNew),'single');
2  U = zeros(size(imNew));
3  V = zeros(size(imNew));
4  Ux = zeros(size(imNew),'single');
5  Uy = zeros(size(imNew),'single');
6  Ut = zeros(size(imNew),'single');
7  Vx = zeros(size(imNew),'single');
8  Vy = zeros(size(imNew),'single');
9  Vt = zeros(size(imNew),'single');
10 imPrev = imNew;
11 Uprev = U;
12 Vprev = V;
```

Kode Sumber 4.3 Implementasi Inisialisasi

Pada baris 1 sampai baris 9 secara berurutan merupakan inisialisasi fitur It, U, V, Ux, Uy, Ut, Vx, Vy, dan Vt. Sedangkan baris 10, 11, dan 12 merupakan inisialisasi citra video serta fitur u dan v pada *frame* sebelumnya.

4.3.3 Perhitungan Fitur I_t, u, dan v

Fitur *optical flow* selanjutnya adalah fitur It, u, dan v. It adalah turunan parsial pertama dari intensitas $I(x, y, t)$ terhadap *frame* t. u dan v komponen *optical flow* yang merupakan perpindahan posisi piksel (x, y) dengan intensitas I berdasarkan t. Implementai perhitungan fitur It, u, dan v ditunjukkan pada Kode Sumber 4.4.

```
1  gg = [0.2163,0.5674,0.2163];
2  tInt = 0.5;
3  It = tInt*It + (1-tInt)*2*conv2(gg,gg,imNew
   - imPrev,'same');
4  for i = 1:MaxIterations
5      uAvg=conv2(U,kern,'same');
6      vAvg=conv2(V,kern,'same');
7      U = uAvg - dx.*(dx.*uAvg + dy.*vAvg  +
   dt)./(eta.^2 + dx.^2 +dy.^2);
8      V = vAvg - dy.*(dx.*uAvg + dy.*vAvg  +
   dt)./(eta.^2 + dx.^2 +dy.^2);
9  end
```

Kode Sumber 4.4 Perhitungan Fitur I_t, u, dan v

Perhitungan fitur It diunjukkan pada baris 3, perhitungan fitur u dan v ditunjukkan pada baris 4 sampai baris 9 mengacu pada metode *Horn and Schunk* pada persamaan (2.4) dan (2.5).

4.3.4 Perhitungan Fitur u_t dan v_t

Fitur optical flow selanjutnya adalah u_t dan v_t yang merupakan turunan parsial pertama fitur u dan v terhadap t.

Implementasi perhitungan fitur u_t dan v_t ditunjukkan pada Kode Sumber 4.5. Perhitungan u_t dan v_t masing-masing ditunjukkan pada baris 3 dan 4.

```
1  gg = [0.2163,0.5674,0.2163];
2  tInt = 0.5;
3  Ut = tInt*ut + (1-tInt)*2*conv2(gg,gg,U -
   Uprev,'same');
4  Vt = tInt*vt + (1-tInt)*2*conv2(gg,gg,V -
   Vprev,'same');
```

Kode Sumber 4.5 Implementasi Perhitungan Fitur u_t dan v_t

4.3.1 Perhitungan Fitur *Div, Vor, Gten,* dan *Sten*

Fitur optical flow selanjutnya adalah tur Div, Vor, Gten, dan Sten. *Div* (*divergence*) adalah perbedaan spasial dari *flow field* yang ada pada setiap posisi piksel. *Vor* (*voricity*) digunakan untuk mengukur putaran lokal di sekitar sumbu yang tegak lurus terhadap bidang *flow field*. *Gten* dan *Sten* adalah invarian tensor yang tetap konstan tidak peduli sistem koordinat apa yang mereka rujuk. Implementasi fitur-fitur ini ditunjukkan pada Kode Sumber 4.6.

```
1  gg = [0.2163,0.5674,0.2163];
2  tInt = 0.5;
3  fU = U + Uprev;
4  Ux = tInt*ux + (1-tInt)*conv2(fU(:,[2:end
   end ]) - fU(:,[1 1:(end-1)]),gg','same');
5  Uy = tInt*uy + (1-tInt)*conv2(fU([2:end
   end],: ) - fU([1 1:(end-1)],:),gg,'same');
6  fV = V + Vprev;
7  Vx = tInt*vx + (1-tInt)*conv2(fV(:,[2:end
   end ]) - fV(:,[1 1:(end-1)]),gg','same');
8  Vy = tInt*vy + (1-tInt)*conv2(fV([2:end
   end],: ) - fV([1 1:(end-1)],:),gg,'same');
9  Div = Ux + Vy;
10 Vor = Vx - Uy;
```

```
11  Gten = 0.5*(((Ux+Vy).*(Ux+Vy)) - ((Ux.*Ux)
    + (Uy.*Vx) + (Vx.*Uy) + (Vy.*Vy)));
12  Sten = 0.5*(((Ux+Vy).*(Ux+Vy)) - ((Ux.*Ux)
    + ((Uy+Vx).*(Uy+Vx)./4) +
    ((Uy+Vx).*(Uy+Vx)./4) + (Vy.*Vy)));
```

Kode Sumber 4.6 Implementasi Perhitungan Fitur *Div, Vor, Gten,* dan *Sten*

Seperti pada persamaan (2.8), (2.9), (2.10), dan (2.11), perhitungan Div, Vor, Gten, dan Sten membutuhkan fitur u_x, u_y, v_x, dan v_y yang perhitungannya ditunjukkan pada baris 4, 5, 7, dan 8. Dilanjutkan dengan perhitungan fitur Div, Vor, Gten, dan Sten pada baris 9, 10, 11, dan 12.

4.3.2 Penggabungan 12 Fitur Menjadi Matriks Fn

Setelah mendapat 12 fitur *optical flow* dilakukanlah penggabungan fitur-fitur tersebut ke dalah sebuah matriks Fn. Implementasi penggabungan 12 fitur ini ditunjukkan pada Kode Sumber 4.7.

```
1   temp(1,:) = reshape(x,1,[]);
2   temp(2,:) = reshape(y,1,[]);
3   temp(3,:) = reshape(t,1,[]);
4   temp(4,:) = reshape(u,1,[]);
5   temp(5,:) = reshape(v,1,[]);
6   temp(6,:) = reshape(It,1,[]);
7   temp(7,:) = reshape(Ut,1,[]);
8   temp(8,:) = reshape(Vt,1,[]);
9   temp(9,:) = reshape(Div,1,[]);
10  temp(10,:) = reshape(Vor,1,[]);
11  temp(11,:) = reshape(Gten,1,[]);
12  temp(12,:) = reshape(Sten,1,[]);
13  Fn = [Fn temp];
```

Kode Sumber 4.7 Implementasi Penggabungan 12 Fitur Menjadi Matriks Fn

Baris 1 sampai baris 12 menunjukkan perubahan matriks masing-masing fitur *optical flow* menjadi vektor baris. Selanjutnya vektor-vektor baris tersebut disatukan menjadi matriks Fn ditunjukkan pada baris 13. Proses ini dilakukan sebanyak L untuk menggabungkan seluruh fitur citra-citra video berurutan dalam satu segmen sebelum masuk ke proses selanjutnya.

4.3.3 Normalisasi

Setelah iterasi sebanyak L, terkumpulah *bag of feature vector* pada matriks Fn. Sebelum dilakukannya perhitungan matriks kovariannya, terlebih dahulu fitur-fitur optical flow tersebut dinormalisasi agar terdistribusi merata. Implementasi normalisasi ditunjukkan pada Kode Sumber 4.8.

```
1  [m,n] = size(Fn);
2  for i = 1 : m
3      Fn(i,:)= normalize(Fn(i,:),'range');
4  end
```

Kode Sumber 4.8 Implementasi Normalisasi

Baris pertama menunjukan pengambilan ukuran baris matriks Fn. Selanjutnya pada baris 2 sampai baris 4 dilakukan normalisasi skala pada setiap fitur *optical flow* pada matriks Fn sebanyak jumlah row matriks Fn yang merupakan banyaknya fitur yang dikumpulkan yaitu 12.

4.4 Implementasi Perhitungan Matriks Kovarian

Proses selanjutnya setelah *bag of feature vector* pada matriks Fn dinormalisasi, dilakukanlah perhitungan matriks kovariannya untuk mengurangi jumlah fitur yang sangat banyak. Implementasi perhitungan matriks kovarian ditunjukkan pada Kode Sumber 4.9.

```
1  [m,n] = size(Fn);
2  Fnmean = mean(Fn,2);
3  Fnmin = Fn - Fnmean;
4  Cov = Fnmin * Fnmin.'*1/n;
```

Kode Sumber 4.9 Implementasi Perhitungan Matriks Kovarian

Baris pertama menunjukan pengambilan ukuran baris matriks Fn. Baris 2 menunjukan perhitungan mean matriks Fn terhadap masing-masing fitur. Selanjutnya pada baris 3 dan 4 adalah perhitungan matriks kovarian dari matriks Fn sesuai dengan persamaan (2.12).

4.5 Implementasi Perhitungan Matriks Log Kovarian

Matriks kovarian dengan ukuran tertentu tidaklah membentuk suatu ruang vector karena tidak diikuti oleh perkalian dengan *scalar* negatif, sehingga membentuk "*closed convex cone*" yang mana kebanyakan dari algoritma *machine learning* bekerja dengan fitur yang berada pada ruang Euclidean, bukan *convex cone*. Oleh sebab itu dibutuhkanlah perhitungan matriks log kovarian untuk memetakan bentuk *convex cone* pada matriks kovarian ke dalam ruang vector. Implementasi perhitungan matriks log kovarian ditunjukkan pada Kode Sumber 4.10.

```
1  [V,D] = eig(Cov);
2  Ddiag = diag(D);
3  Dlog = log(Ddiag);
4  D = diag(Dlog);
5  LogCov = V*D*V.';
```

Kode Sumber 4.10 Implementasi Perhitungan Matriks Log Kovarian

Baris 1 menunjukkan perhitungan untuk mendapatkan *orthonormal eigenvectors* dan *eigenvalues* matriks kovarian. Mengacu pada persamaan (2.16), perhitungan matriks diagonal dari matriks *eigevalues* dengan mengganti entri diagonal matriks *D* dengan nilai logaritmanya ditunjukkan pada baris 2 sampai baris 4,

dilanjutkan dengan perhitungan matriks log kovarian pada baris 5. Matriks log kovarian inilah yang akan merepresentasikan suatu aktivitas.

4.6 Implementasi Klasifikasi Aktivitas

Proses selanjutnya adalah klasifikasi data testing untuk menentukan jenis aktivitas yang dilakukan. Implementasi klasifikasi aktivitas dengan metode NN classifier ditunjukkan pada Kode Sumber 4.11. Baris 1 menunjukkan pembuatan model data *training* menggukan jarak euclidean sebagai *distance metric*, dilanjutkan dengan prediksi kelas aktivitas pada baris 2.

```
1  mdl = fitcknn(train,train_label,'Distance',
   'euclidean');
2  test_label = predict(mdl,test);
```

Kode Sumber 4.11 Implementasi Klasifikasi Aktivitas

4.7 Implementasi Pengukuran Kinerja Sistem

Untuk mengukur kinerja sistem pengenalan aktivitas pada video, digunakanlah metode pengukuran 10-*fold cross validation* pada saat klasifikasi aktivitas dengan metode NN *classifier*. Implementasi pengukuran kinerja sistem ditunjukkan pada Kode Sumber 4.12.

```
1  X = xlsread(excelFileName,data);
2  Y = xlsread(excelFileName,dataLabel);
3  mdl = fitcknn(X,Y,'Distance','euclidean');
4  cvmdl = crossval(mdl);
5  cvmdl_predict = kfoldPredict(cvmdl);
6  cvmdl_conf =
   confusionmat(cvmdl.Y,cvmdl_predict);
7  validationAccuracy = 1 - kfoldLoss(cvmdl,
   'LossFun', 'ClassifError');
```

Kode Sumber 4.12 Implementasi Pengukuran Kinerja Sistem

Baris 1 dan baris 2 masing-masing menunjukan pembacaan data uji dan data label klasifikasinya yang telah disimpan sebelumnya dalam file excel. Selanjutnya dilakukan pembentukkan model dengan metode NN *classifier* pada baris 3. Implementasi uji coba menggunakan *10-fold cross validation* ditunjukkan pada baris 4. Selanjutnya untuk mendapatkan data label prediksi ditunjukkan pada baris 5, untuk perhitungan *confusion matrix* pada baris 6, dan yang terakhir perhitungan akurasi ditunjukkan pada baris 7.

BAB V
UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan

5.1 Lingkungan Uji Coba

Lingkungan uji coba pada tugas akhir ini adalah sebuah personal computer (PC). Spesifikasi PC dari sisi perangkat keras adalah memiliki prosesor Intel Core i3 3240 dengan kecepatan 3,40 GHz dan memori untuk proses sebesar 4,00 GB. PC yang digunakan memiliki sistem operasi Windows 10. Pada sisi perangkat lunak, uji coba pada tugas akhir ini dilakukan dengan menggunakan software MATLAB R2018a dan Microsoft Excel.

5.2 Data Uji Coba

Data uji coba yang digunakan sebagai data masukan ada dua macam yaitu dataset Weizmann dan dataset CCTV. Berikut penjelasan mengenai masing-masing dataset.

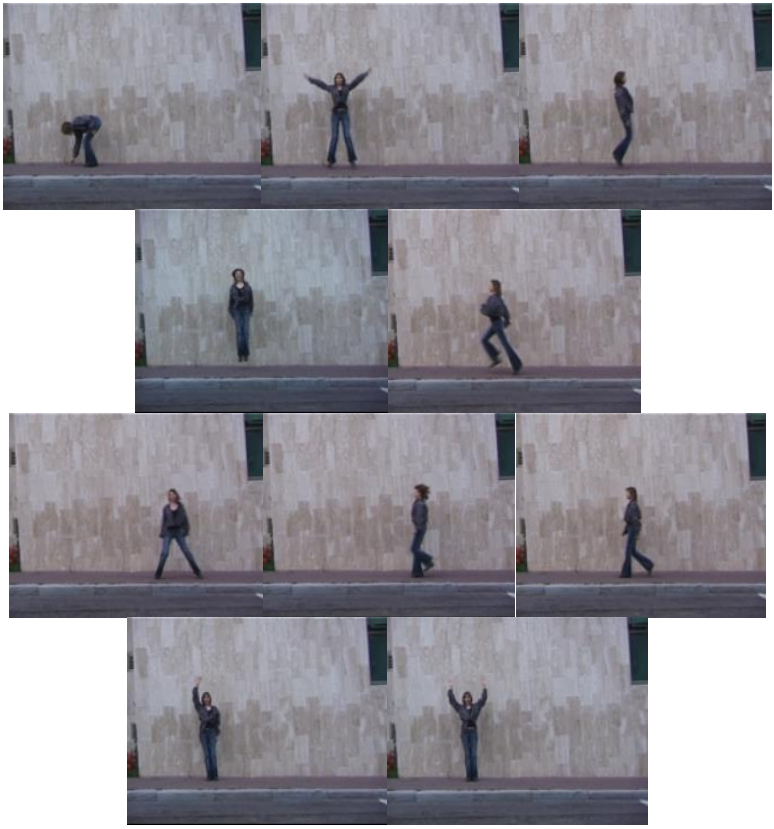
5.2.1 Dataset Weizmann

Dataset Weizmann merupakan video berukuran 144 x 180 piksel dengan *framerate* 25 *fps* dan citra warna RGB. Dataset ini memiliki 10 jenis aktivitas yaitu *run, walk, wave2, jump, pjump, jack, side, skip, wave1*, dan *bend*. Masing-masing aktivitas dilakukan oleh sembilan orang yang berbeda. Spesifikasi dataset ini ditunjukkan pada Tabel 5.1. Contoh citra video dataset ini ditunjukkan pada Gambar 5.1.

Tabel 5.1 Spesifikasi Dataset Weizmann

Keterangan		Spesifikasi
Ukuran Resolusi		180 x 144 piksel
Frame Rate		25 *fps*
Ekstensi		.avi
Jumlah Video		93
Jumlah Orang		9
Jumlah Aktivitas		10
Aktivitas		*Run, walk, wave2, jump, pjump, jack, side, skip, wave1, dan bend.*
Jumlah Video per Aktivitas	*Run*	10
	Walk	10
	Wave2	9
	Jump	9
	Pjump	9
	Jack	9
	Side	9
	Skip	10
	Wave1	9
	Bend	9
Durasi Video	*Run*	1 – 2 detik
	Walk	1 – 3 detik
	Wave2	2 – 4 detik
	Jump	1 – 2 detik
	Pjump	1 – 5 detik
	Jack	2 – 5 detik
	Side	1 – 2 detik
	Skip	1 – 2 detik
	Wave1	2 – 5 detik
	Bend	2 – 3 detik
Ukuran File	*Run*	2737 – 5851 KB
	Walk	3269 – 9194 KB
	Wave2	3877 – 8662 KB

Keterangan		Spesifikasi
	Jump	2889 – 5472 KB
	Pjump	3193 – 9650 KB
	Jack	3877 – 11093 KB
	Side	2965 – 4864 KB
	Skip	2813 – 7067 KB
	Wave1	4028 – 9498 KB
	Bend	4636 – 7067 KB



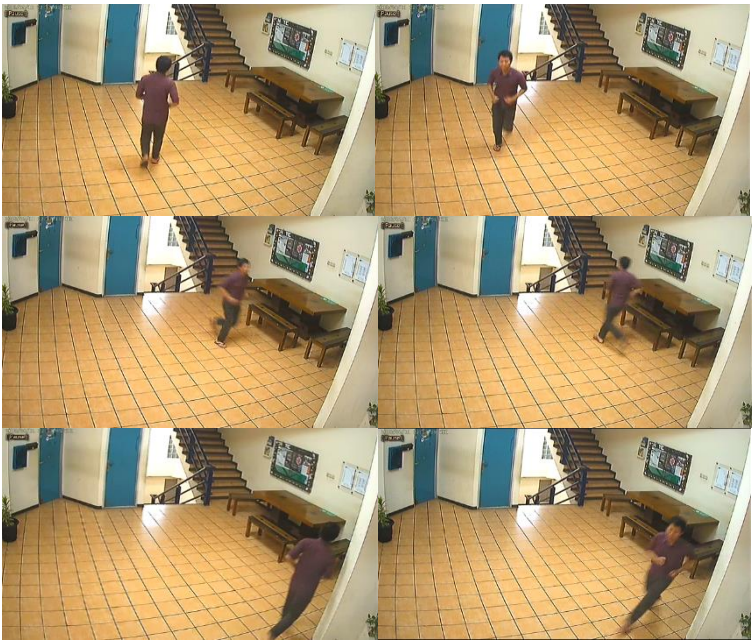
Gambar 5.1 Contoh Citra Video pada Dataset Weizmann
(*Bend, Jack, Jump, Pjump, Run, Side, Skip, Walk, Wave1, Wave2*)

5.2.2 Dataset CCTV

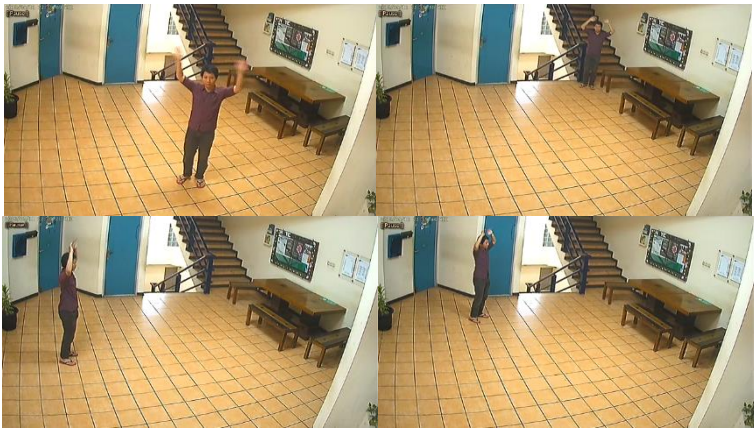
Dataset CCTV merupakan video berukuran 1280 x 720 piksel dengan *framerate* 25 *fps* dan citra warna RGB. Dataset ini memiliki tiga jenis aktivitas yaitu berlari, berjalan, dan melambaikan kedua tangan. Masing-masing aktivitas dilakukan oleh enam orang yang berbeda. Spesifikasi dataset ini ditunjukkan pada Tabel 5.2. Contoh citra video dataset ini ditunjukkan pada Gambar 5.2, 5.3, 5.4, dan 5.5.

Tabel 5.2 Spesifikasi Dataset CCTV

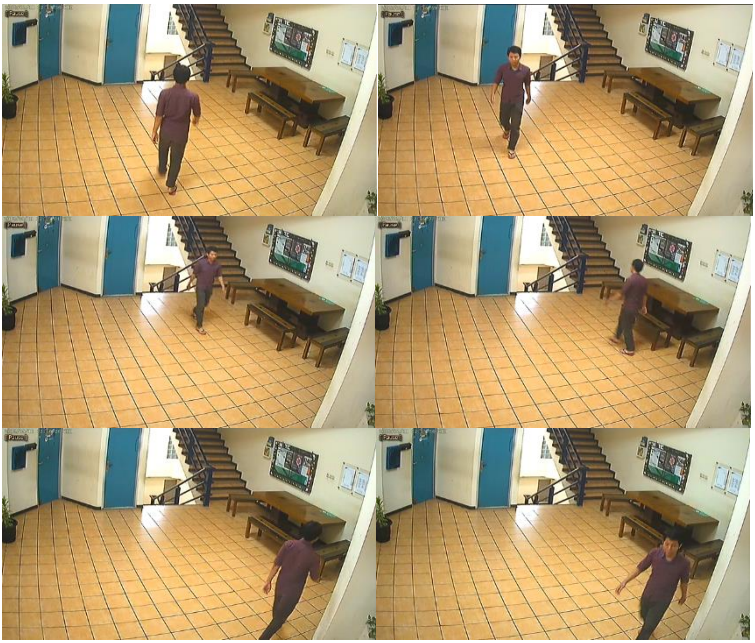
Keterangan		Spesifikasi
Ukuran Resolusi		1280 x 720 piksel
Frame Rate		25 *fps*
Ekstensi		.avi
Jumlah Video		288
Jumlah Orang		6
Waktu		Pagi (08.00-10.00) Siang (13.00-14.00) Malam (20.00-22.00)
Jumlah Aktivitas		3
Aktivitas		6 berlari, 6 berjalan, 4 melambai (16)
Jumlah Video per Waktu		96
Jumlah Video per Aktivitas	Berlari	108
	Berjalan	108
	Melambai	72
Durasi Video	Berlari	2 – 10 detik
	Berjalan	1 – 7 detik
	Melambai	1 – 10 detik
Ukuran File	Berlari	197 – 641 KB
	Berjalan	166 – 478 KB
	Melambai	85 – 527 KB



Gambar 5.2 Contoh Citra Video pada Dataset CCTV – Berlari



Gambar 5.3 Contoh Citra Video pada Dataset CCTV – Melambaikan Kedua Tangan



Gambar 5.4 Contoh Citra Video pada Dataset CCTV – Berjalan

Gambar 5.5 Contoh Citra Video pada Dataset CCTV – Pagi, Siang, Malam

5.3 Skenario Uji Coba

Uji coba dilakukan untuk mengetahui nilai-nilai parameter yang tepat untuk digunakan pada masing-masing proses. Nilai parameter yang tepat penting untuk diketahui karena penggunaan parameter yang tepat akan memberikan hasil yang terbaik pada keluaran tiap proses. Selain menentukan nilai parameter, uji coba berguna melihat performa sistem dengan metode lainnya. Skenario pengujian dilakukan pada dua jenis data yaitu pada dataset weizmann dan data CCTV.

Skenario uji coba pada dataset Weizmann ialah sebagai berikut.

1. Uji Coba dengan parameter panjang *frame L*
2. Uji Coba dengan parameter *overlapping frame P*

Skenario uji coba pada data CCTV ialah sebagai berikut.

1. Uji Coba dengan parameter panjang *frame L*
2. Uji Coba dengan parameter *overlapping frame P*
3. Uji Coba dengan parameter waktu pengambilan data CCTV

5.4 Skenario Uji Coba pada Dataset Weizmann

Uji coba pada dataset weizmann dilakukan untuk mengetahui kesesuaian hasil akurasi dengan yang sudah diimplementasikan pada paper acuan. Uji coba ini dilakukan menggunakan dua parameter yaitu panjang *frame L* dan *overlapping frame* ialah sebagai berikut.

5.4.1 Skenario Uji Coba Panjang *Frame L*

Skenario uji coba panjang *frame L* dilakukan untuk mengetahui seberapa baik panjang *frame* dapat mempengaruhi hasil pengenalan aktivitas manusia pada dataset weizmann. Uji coba dilakukan dengan melakukan variasi pada panjang *frame L* yang digunakan dalam satu segmen, yaitu L = 8 dan L = 20 mengacu pada riset yang telah dilakukan sebelumnya [1]. Uji coba

ini dilakukan pada sistem menggunakan metode 10-*fold cross validation*.

Hasil uji coba tertera pada Tabel 5.3 yang menunjukkan hasil *accuracy, recall,* dan *precision* pada masing-masing kelas aktivitas di setiap skenario. *Confussion matrix* untuk uji coba skenario ini terdapat pada lampiran.

Dari tabel tersebut dapat dilihat bahwa panjang *frame L* = 20 memberikan hasil yang terbaik untuk pengenalan aktivitas manusia pada dataset weizmann dengan nilai *accuracy* sebesar 95.51%, rata-rata *recall* sebesar 94.32%, dan rata-rata *precision* sebesar 95.02%. Untuk *recall* kelas akivitas *run, walk, wave2, jump, pjump, jack, side, skip, wave1, dan bend* pada skenario panjang *frame L* = 20, masing-masing adalah 94.44%, 99.24%, 97.35%, 84.51%, 100%, 100%, 82.61%, 91.78%, 96.69%, dan 96.58%. Sedangkan untuk *specifity* kelas aktivitas *run, walk, wave2, jump, pjump, jack, side, skip, wave1, dan bend* pada skenario panjang *frame L* = 20, masing-masing adalah 89.47%, 90.91%, 94.83%, 96.77%, 97.89%, 100%, 93.44%, 89.33%, 97.50%, dan 100%.

Tabel 5.3 Hasil Uji Coba Sistem pada Dataset Weizmann dengan Panjang *Frame L* = 8, 20

(%)		Panjang *Frame*	
		L = 8	L = 20
Accuracy		73.39	**95.51**
Recall	*Run*	67.86	94.44
	Walk	80.75	99.24
	Wave2	75.71	97.35
	Jump	51.02	84.51
	Pjump	72.50	**100.00**
	Jack	95.76	**100.00**
	Side	42.71	82.61
	Skip	46.60	91.78
	Wave1	75.68	96.69
	Bend	93.75	96.58
Rata-Rata *Recall*		70.23	94.32
Precision	*Run*	60.64	89.47
	Walk	75.58	90.91
	Wave2	69.74	94.83
	Jump	56.18	96.77
	Pjump	85.29	97.89
	Jack	86.34	**100.00**
	Side	50.62	93.44
	Skip	45.71	89.33
	Wave1	81.16	97.50
	Bend	94.41	**100.00**
Rata-Rata *Precision*		70.57	**95.02**

5.4.2 Skenario Uji Coba *Overlapping Frame P*

Skenario uji coba *overlapping frame P* dilakukan untuk mengetahui seberapa baik *overlapping frame* dapat mempengaruhi hasil pengenalan aktivitas manusia pada dataset weizmann. Uji coba dilakukan dengan melakukan variasi pada *overlapping frame P* yang digunakan antar segmen, yaitu *P* = 1, 2, 4, dan 6 untuk mengetahui pengaruh setiap penambahan dua frame pada *overlapping frame*. Uji coba ini dilakukan pada sistem menggunakan metode 10-fold cross validation.

Hasil uji coba tertera pada Tabel 5.4 yang menunjukkan hasil *accuracy, recall*, dan *precision* pada masing-masing kelas aktivitas di setiap skenario. *Confussion matrix* untuk uji coba skenario ini terdapat pada lampiran.

Dari tabel tersebut dapat dilihat bahwa panjang *overlapping frame P* = 1 memberikan hasil yang terbaik untuk pengenalan aktivitas manusia pada dataset weizmann dengan nilai *accuracy* sebesar 99.31%, rata-rata *recall* sebesar 99.19%, dan rata-rata *precision* sebesar 99.21%. Untuk *recall* kelas akivitas *run, walk, wave2, jump, pjump, jack, side, skip, wave1, dan bend* pada skenario panjang *overlapping frame P* = 1, masing-masing adalah 98.60%, 99.42%, 99.78%, 98.61%, 100%, 99.82%, 98.17%, 98.99%, 99.38%, dan 99.15%. Sedangkan untuk *specifity* kelas aktivitas *run, walk, wave2, jump, pjump, jack, side, skip, wave1, dan bend* pada skenario *overlapping frame P* = 1, masing-masing adalah 99.07%, 99.42%, 98.91%, 99.30%, 100%, 99.82%, 98.53%, 97.67%, 99.58%, dan 99.78%. Selain itu, dapat dilihat bahwa semakin tinggi *overlapping frame*, semakin banyak pula data yang dihasilkan yaitu 3920 data segmen sebagai hasil data terbanyak dengan waktu pembentukan model klasifikasi selama 20.21 detik.

Tabel 5.4 Hasil Uji Coba Sistem pada Dataset Weizmann dengan *Overlaping Frame P* = 1, 2, 4, 6

(%)		*Overlapping Frame*			
		P = 1	P = 2	P = 4	P = 6
Jumlah Data		3920	1960	980	654
Waktu Pembentukan Model Klasifikasi		20.21 *s*	18.42 *s*	23.92 *s*	19.62 *s*
Accuracy		**99.31**	98.21	95.51	92.96
Recall	*Run*	98.60	99.07	94.44	91.89
	Walk	99.42	98.08	99.24	96.51
	Wave2	99.78	99.56	97.35	94.67
	Jump	98.61	95.83	84.51	79.17
	Pjump	**100.00**	100.00	100.00	96.72
	Jack	99.82	100.00	100.00	98.92
	Side	98.17	93.43	82.61	84.78
	Skip	98.99	96.60	91.78	81.63
	Wave1	99.38	98.76	96.69	96.25
	Bend	99.15	97.86	96.58	94.87
Rata-Rata *Recall*		**99.19**	97.92	94.32	91.54
Precision	*Run*	99.07	95.54	89.47	89.47
	Walk	99.42	97.70	90.91	92.22
	Wave2	98.91	98.25	94.83	95.95
	Jump	99.30	96.50	96.77	92.68
	Pjump	**100.00**	100.00	97.89	93.65
	Jack	99.82	99.64	100.00	96.84
	Side	98.53	96.24	93.44	82.98
	Skip	97.67	96.60	89.33	80.00
	Wave1	99.58	98.35	97.50	96.25
	Bend	99.78	100.00	100.00	98.67
Rata-Rata *Precision*		**99.21**	97.88	95.02	91.87

5.5 Skenario Uji Coba pada Dataset CCTV

Uji coba pada dataset CCTV dilakukan menggunakan tiga parameter yaitu panjang *frame L*, *overlapping frame P*, dan waktu pengambilan data ialah sebagai berikut.

5.5.1 Skenario Uji Coba Panjang *Frame L*

Waktu yang dibutuhkan untuk merepresentasikan aktivitas manusia adalah sekitar 0.4-0.8 detik [1]. *Framerate* pada video CCTV adalah sebesar 25 *fps*. Dengan kata lain, waktu yang dibutuhkan untuk merepresentasikan suatu aktivitas dalam dataset CCTV adalah sebesar 10 – 20 *frame*. Oleh karena itu dilakukan pengujian pada panjang *frame L* sebesar 10, 15, dan 20.

Hasil uji coba sistem pada dataset CCTV dengan varian panjang *frame L* ditunjukkan pada Tabel 5.5 yang menunjukkan hasil akurasi, *recall,* dan *precision* pada masing-masing kelas aktivitas di setiap skenario. Pengujian ini dilakukan untuk mengetahui seberapa baik panjang *frame* dapat mempengaruhi hasil pengenalan aktivitas manusia pada dataset CCTV serta untuk mengetahui panjang *frame* yang paling baik dalam merepresentasikan suatu aktivitas pada dataset CCTV. *Confussion matrix* untuk uji coba skenario ini terdapat pada lampiran.

Dari tabel tersebut dapat dilihat bahwa panjang *frame L* = 15 memberikan hasil yang terbaik untuk pengenalan aktivitas manusia pada dataset CCTV dengan nilai *accuracy* sebesar 79,11%, rata-rata *recall* sebesar 75.85%, dan rata-rata *precision* sebesar 76.36%. Hasil *recall* kelas akivitas berlari, berjalan, dan melambaikan kedua tangan, masing-masing adalah 48.77%, 80.50%, dan 98.29%. Sedangkan untuk *precision* kelas aktivitas berlari, berjalan, dan melambaikan kedua tangan, masing-masing adalah 59.50%, 76.83%, 92.74%.

Tabel 5.5 Hasil Uji Coba Sistem pada Dataset CCTV dengan Panjang *Frame L* = 10, 15, 20

(%)		Panjang *Frame*		
		L = 10	L = 15	L = 20
Accuracy		65.33	**79.11**	74.40
Recall	Berlari	39.10	48.77	40.20
	Berjalan	63.19	80.50	74.37
	Melambai	88.19	**98.29**	95.64
Rata-Rata *Recall*		63.49	**75.85**	70.07
Precision	Berlari	44.49	59.50	48.48
	Berjalan	66.73	76.83	74.88
	Melambai	75.68	**92.74**	85.75
Rata-Rata *Precision*		62.30	**76.36**	69.71

5.5.2 Skenario Uji Coba *Overlapping Frame P*

Skenario uji coba *overlapping frame P* dilakukan untuk mengetahui seberapa baik *overlapping frame* dapat mempengaruhi hasil pengenalan aktivitas manusia pada dataset weizmann. Uji coba dilakukan dengan melakukan variasi pada *overlapping frame P* yang digunakan antar segmen, yaitu *P* = 1, 2, 4, dan 6 untuk mengetahui pengaruh setiap penambahan dua frame pada *overlapping frame*. Uji coba ini dilakukan pada sistem menggunakan metode 10-fold cross validation.

Hasil uji coba tertera pada Tabel 5.6 yang menunjukkan hasil *accuracy*, *recall*, dan *precision* pada masing-masing kelas aktivitas di setiap skenario. *Confussion matrix* untuk uji coba skenario ini terdapat pada lampiran.

Tabel 5.6 Hasil Uji Coba Sistem pada Dataset CCTV dengan *Overlaping Frame P* = 1, 2, 4, 6

(%)		*Overlapping Frame*			
		P = 1	P = 2	P = 4	P = 6
Jumlah Data		4307	2154	1077	718
Waktu Pembentukan Model Klasifikasi		10.07	9.15	9.17	12.66
Accuracy		**95.80**	92.11	79.11	76.32
Recall	Berlari	91.47	83.16	48.77	47.24
	Berjalan	95.23	91.49	80.50	75.39
	Melambai	99.57	99.15	98.29	97.86
Rata-Rata *Recall*		**95.42**	91.27	75.85	73.50
Precision	Berlari	91.10	84.20	59.50	55.00
	Berjalan	95.93	92.16	76.83	75.39
	Melambai	98.87	97.35	92.74	89.11
Rata-Rata *Precision*		**95.30**	91.24	76.36	73.16

Dari tabel tersebut dapat dilihat bahwa *overlapping frame P* = 1 memberikan hasil yang terbaik untuk pengenalan aktivitas manusia pada dataset CCTV dengan nilai *accuracy* sebesar 95,80%, rata-rata *recall* sebesar 95.42%, dan rata-rata *precision* sebesar 95.30%. Hasil *recall* kelas akivitas berlari, berjalan, dan melambaikan kedua tangan, masing-masing adalah 91.47%, 95.23%, dan 99.57%. Sedangkan untuk *precision* kelas aktivitas berlari, berjalan, dan melambaikan kedua tangan, masing-masing adalah 91.10%, 95.93%, 98.30%.

5.5.3 Skenario Uji Coba Waktu Pengambilan Data

Uji coba sistem pada dataset CCTV juga dilakukan dengan varian waktu pengambilan data yaitu saat pagi, siang, dan malam

hari. Pengujian ini dilakukan untuk mengetahui seberapa baik intensitas cahaya pada waktu-waktu tersebut dapat mempengaruhi hasil pengenalan aktivitas manusia pada dataset CCTV.

Hasil uji coba sistem pada dataset CCTV dengan varian waktu pengambilan data ditunjukkan pada Tabel 5.7 yang menunjukkan hasil akurasi, *recall*, dan *precision* pada masing-masing kelas aktivitas. *Confussion matrix* untuk uji coba skenario ini terdapat pada lampiran.

Dari tabel tersebut dapat dilihat bahwa waktu pengambilan data pada malam hari memberikan hasil yang terbaik untuk pengenalan aktivitas manusia pada dataset CCTV dengan nilai *accuracy* sebesar 95,86%, rata-rata *recall* sebesar 96.29%, dan rata-rata *precision* sebesar 96.01%. Hasil *recall* kelas akivitas berlari, berjalan, dan melambaikan kedua tangan pada skenario waktu pengambilan data pada malam hari, masing-masing adalah 93.57%, 96.39%, 98.91%, dan 99.57%. Sedangkan untuk *precision* kelas aktivitas berlari, berjalan, dan melambaikan kedua tangan pada skenario waktu pengambilan data pada malam hari, masing-masing adalah 94.60%, 96.24%, 97.20%.

Tabel 5.7 Hasil Uji Coba Sistem pada Dataset CCTV dengan Waktu Pagi, Siang, dan Malam

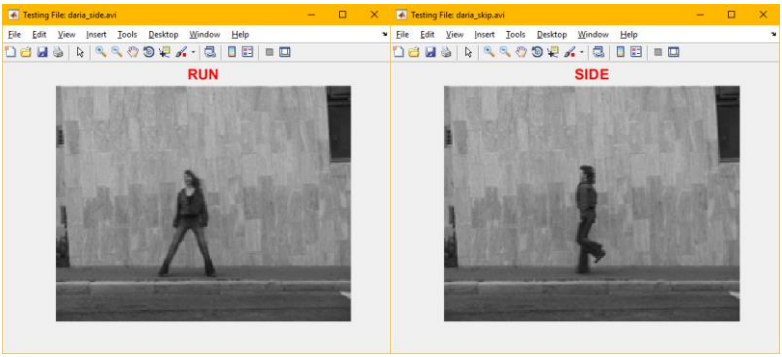
(%)		Waktu Pengambilan Data		
		Pagi	Siang	Malam
Accuracy		91.99	95.80	**95.86**
Recall	Berlari	83.64	91.47	93.57
	Berjalan	92.42	95.23	96.39
	Melambai	97.88	99.57	98.91
Rata-Rata *Recall*		91.31	95.42	**96.29**
Precision	Berlari	87.43	91.10	94.60
	Berjalan	92.14	95.93	96.24
	Melambai	95.10	98.87	97.20
Rata-Rata *Precision*		91.56	95.30	**96.01**

5.6 Evaluasi Uji Coba pada Dataset Weizmann

Evaluasi uji coba pada dataset weizmann yang dilakukan dengan menggunakan dua parameter yaitu panjang *frame L* dan *overlapping frame P* ialah sebagai berikut.

5.6.1 Evaluasi Uji Coba Panjang *Frame L*

Berdasarkan hasil uji coba sistem pada dataset weizmann, diketahui bahwa penggunaan panjang *frame L* = 20 oleh sistem pengenalan aktivitas pada video ini menghasilkan *accuracy, recall*, dan *precision* yang lebih baik dibandingkan penggunaan panjang *frame L* = 8. Selain itu, diketahui bahwa kelas aktivitas *pjump, jack*, dan *bend* menghasilkan *recall* dan *precision* yang paling baik dari tujuh kelas aktivitas lainnya. Hal ini menunjukkan bahwa kelas aktivitas *pjump, jack*, dan *bend* memiliki karakteristik yang berbeda dibandingkan kelas aktivitas lainnya. Karakteristik tersebut dapat dilihat dari pola gerakan yang dihasilkan oleh aktivitas *pjump, jack*, dan *bend* yang dilakukan yaitu bergerak hanya pada satu titik di layar video, dimana aktivitas lainnya seperti *run, walk, jump, side,* dan *skip* dilakukan bergerak dari sisi kanan ke sisi kiri maupun sebaliknya pada layar video. Sedangkan bila dibandingkan dengan kelas aktivitas lainnya yang sama-sama bergerak pada satu titik seperti *wave2* dan *wave1,* gerakan *pjump, jack*, dan *bend* memiliki pergerakan yang berbeda-beda, dimana *wave2* dan *wave1* sama-sama memiliki unsur pergerakan lambaian tangan. Perbandingan gerakan juga terletak pada bagian yang bergerak dimana *pjump, jack*, dan *bend* menghasilkan pergerakan piksel yang lebih banyak dibandingkan dengan *wave2* dan *wave1*. Berdasarkan hasil uji coba pada panjang *frame L* = 8, hasil klasifikasi terburuk terjadi pada kelas aktivitas *side* dan *skip.* Contoh hasil misklasifikasi aktivitas tersebut ditunjukkan pada Gambar 5.6.

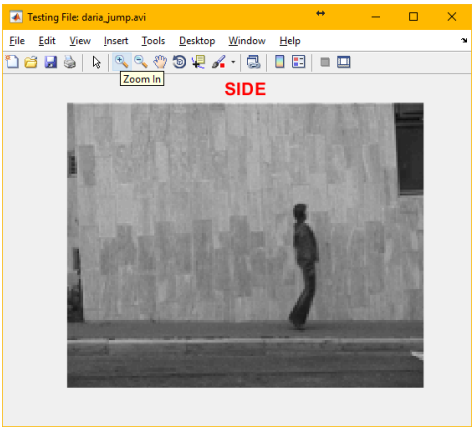


Gambar 5.6 Contoh Misklasifikasi Kelas Aktivitas *Side* dan *Skip* pada Penggunaan Panjang *Frame L* = 8

5.6.2 Evaluasi Uji Coba *Overlapping Frame P*

Berdasarkan hasil uji coba sistem pada dataset weizmann dengan mengimplementasikan panjang *frame* terbaik yang didapat dari uji coba sebelumnya, diketahui bahwa penggunaan *overlapping frame P* = 1 oleh sistem pengenalan aktivitas pada video ini menghasilkan *accuracy*, *recall*, dan *precision* yang lebih baik dibandingkan penggunaan *overlapping frame P* = 2, 4 maupun P = 6. Namun, dapat dilihat pula bahwa penggunaan *overlapping frame P* = 2, 4, dan 6 juga masih menghasilkan *accuracy*, *recall*, dan *precision* yang baik. Hal ini membuktikan bahwa penggunaan fitur matriks kovarian mampu mereduksi jumlah fitur pada menjadi jauh lebih kecil serta mampu merepresentasikannya sebagai suatu aktivitas. Selain itu, diketahui bahwa semakin kecil *overlapping frame* semakin banyak pula jumlah data yang dihasilkan, namun tidak memengaruhi waktu pembentukan model klasifikasi, dilihat dari waktu yang diperlukan dalam pembentukan model klasifikasi yang tidak berbanding lurus maupun terbalik terhadap jumlah data yang digunakan. Hal ini menunjukkan bahwa semakin banyak data yang digunakan, semakin baik pula klasifikasi yang dihasilkan. Sehingga dapat dikatakan bahwa *overlapping frame* yang paling baik dalam merepresentasikan

suatu aktivitas pada dataset weizmann adalah *overlapping frame P* = 1 dengan jumlah data yang paling banyak. Berdasarkan hasil uji coba pada *overlapping frame P* = 6, hasil klasifikasi terburuk terjadi pada kelas aktivitas *jump*. Contoh hasil misklasifikasi kelas tersebut ditunjukkan pada Gambar 5.7.



Gambar 5.7 Contoh Misklasifikasi Kelas Aktivitas *Jump* pada Penggunaan *Overlapping Frame P* = 6

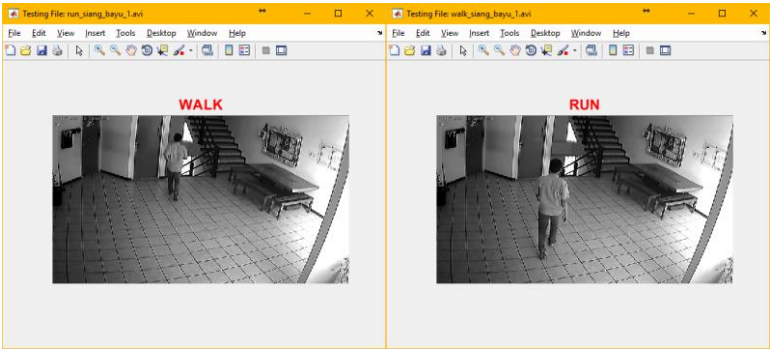
5.7 Evaluasi Uji Coba pada Dataset CCTV

Evaluasi uji coba pada dataset CCTV yang dilakukan dengan menggunakan tiga parameter yaitu panjang *frame L*, *overlapping frame P*, dan waktu pengambilan data ialah sebagai berikut.

5.7.1 Evaluasi Uji Coba Panjang *Frame L*

Berdasarkan hasil uji coba sistem pada dataset CCTV, diketahui bahwa penggunaan panjang *frame L* = 15 oleh sistem pengenalan aktivitas pada video ini menghasilkan *accuracy*, *recall*, dan *precision* yang lebih baik dibandingkan penggunaan panjang *frame L* = 10 maupun L = 20. Sehingga dapat dikatakan bahwa panjang *frame* yang paling baik dalam merepresentasikan suatu

aktivitas pada dataset CCTV adalah panjang *frame L* = 15. Selain itu, diketahui bahwa kelas aktivitas melambai menghasilkan *recall* dan *precision* yang paling baik dari dua kelas aktivitas lainnya. Hal ini menunjukkan bahwa kelas aktivitas melambai memiliki karakteristik yang berbeda dibandingkan kelas aktivitas lainnya. Karakteristik tersebut dapat dilihat dari pola gerakan yang dihasilkan oleh aktivitas melambai yang dilakukan yaitu bergerak hanya pada satu titik di layar video, dimana aktivitas lainnya seperti berlari dan berjalan dilakukan bergerak dari sisi satu ke sisi lainnya pada layar video. Perbandingan gerakan juga terletak pada bagian yang bergerak dimana berlari dan berjalan memiliki kemiripan bentuk pergerakan. Berdasarkan hasil uji coba pada panjang *frame L* = 10, hasil klasifikasi terburuk terjadi pada kelas aktivitas berlari dan berjalan. Contoh hasil misklasifikasi aktivitas tersebut ditunjukkan pada Gambar 5.8.

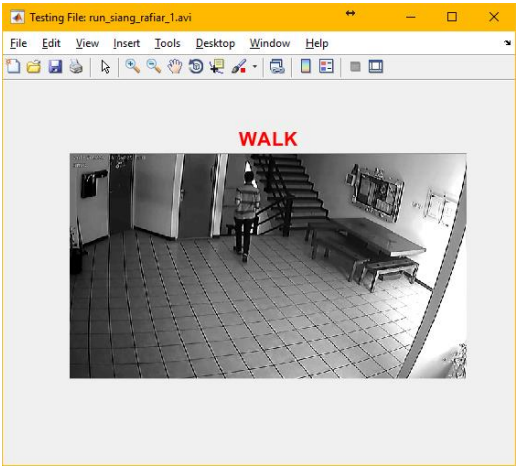


Gambar 5.8 Contoh Misklasifikasi Kelas Aktivitas Berlari dan Berjalan pada Penggunaan Panjang *Frame L* = 10

5.7.2 Evaluasi Uji Coba *Overlapping Frame P*

Berdasarkan hasil uji coba sistem pada dataset CCTV dengan mengimplementasikan panjang *frame* terbaik yang didapat dari uji coba sebelumnya, diketahui bahwa penggunaan *overlapping frame P* = 1 oleh sistem pengenalan aktivitas pada

video ini menghasilkan *accuracy*, *recall*, dan *precision* yang lebih baik dibandingkan penggunaan *overlapping frame P* = 2, 4 maupun P = 6. Dapat dilihat pula bahwa hasil akurasi mulai menurun pada *overlapping frame P* = 4. Selain itu, diketahui bahwa semakin kecil *overlapping frame* semakin banyak pula jumlah data yang dihasilkan, namun tidak memengaruhi waktu pembentukan model klasifikasi, dilihat dari waktu yang diperlukan dalam pembentukan model klasifikasi yang tidak berbanding lurus maupun terbalik terhadap jumlah data yang digunakan.. Hal ini menunjukkan bahwa semakin banyak data yang, semakin baik pula klasifikasi dihasilkan. Sehingga dapat dikatakan bahwa *overlapping frame* yang paling baik dalam merepresentasikan suatu aktivitas pada dataset CCTV adalah *overlapping frame P* = 1 dengan jumlah data yang paling banyak. Berdasarkan hasil uji coba pada *overlapping frame P* = 6, hasil klasifikasi terburuk terjadi pada kelas aktivitas berlari. Contoh hasil misklasifikasi kelas tersebut ditunjukkan pada Gambar 5.9.

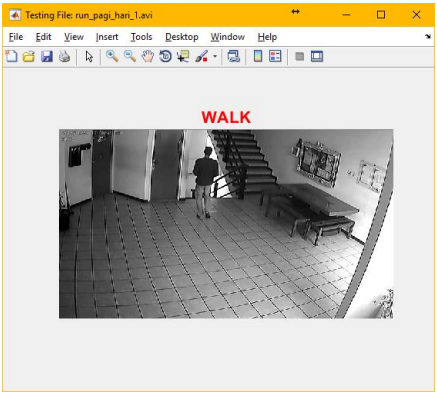


Gambar 5.9 Contoh Misklasifikasi Kelas Aktivitas Berlari pada Penggunaan *Overlapping Frame P* = 6

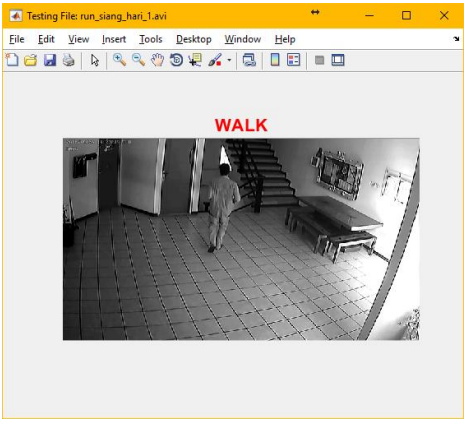
5.7.3 Evaluasi Uji Coba Waktu Pengambilan Data

Berdasarkan hasil uji coba sistem pada dataset CCTV dengan mengimplementasikan panjang *frame* dan *overlapping frame* terbaik yang didapat dari uji coba sebelumnya, diketahui bahwa waktu pengambilan video pada malam hari menghasilkan *accuracy, recall,* dan *precision* yang lebih baik dibandingkan waktu pengambilan lainnya. Sehingga dapat dikatakan bahwa waktu pengambilan data yang paling baik dalam merepresentasikan suatu aktivitas pada dataset CCTV adalah malam hari. Hal ini disebabkan oleh intensitas cahaya pada video yang didapat saat malam hari lebih stabil oleh cahaya lampu sehingga setiap pergerakan yang terjadi memiliki intensitas yang tidak jauh berbeda. Sedangkan intensitas cahaya yang didapat saat pagi dan siang hari dapat dipengaruhi oleh cahaya matahari yang berubah-ubah sesuai dengan pergantian waktu maupun keadaan cuaca.

Namun daripada itu, hasil ujicoba sistem pada dataset CCTV ini dapat dikatakan baik dilihat dari hasil klasifikasi pada setiap waktu dengan nilai rata-rata 90%. Hal ini membuktikan bahwa penggunaan fitur matriks kovarian mampu mereduksi jumlah fitur pada menjadi jauh lebih kecil serta mampu merepresentasikannya sebagai suatu aktivitas. Berdasarkan hasil uji coba waktu pengambilan data pada pagi dan siang hari, hasil klasifikasi terburuk terjadi pada kelas aktivitas berlari. Contoh hasil misklasifikasi kelas tersebut ditunjukkan pada Gambar 5.10 dan 5.11.



Gambar 5.10 Contoh Misklasifikasi Kelas Aktivitas Berlari pada Pagi Hari



Gambar 5.11 Contoh Misklasifikasi Kelas Aktivitas Berlari pada Siang Hari

BAB VI
KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan yang dapat diambil dari hasil uji coba dan perancangan sistem sebagai jawaban dari rumusan masalah yang telah dikemukakan dan saran yang berisi pengembangan yang dapat dilakukan lebih lanjut untuk menyempurnakan sistem pengenalan aktivitas manusia pada data video.

6.1 Kesimpulan

Berikut merupakan kesimpulan yang dapat diambil dari proses pengembangan dan hasil uji coba sistem.

1. Fitur matriks kovarian dapat digunakan untuk mereduksi *bag of feature vector* dari fitur *optical flow* pada data video menjadi jauh lebih kecil namun tetap dapat merepresentasikan suatu aktivitas.
2. Berdasarkan uji coba yang dilakukan, panjang *frame* dan *overlapping frame* yang paling baik dalam merepresentasikan aktivitas pada dataset weizmann adalah panjang *frame L* = 20 dan *overlapping frame P* = 1 dengan nilai *accuracy* sebesar 99.31%, rata-rata *recall* sebesar 99.19%, dan rata-rata *precision* sebesar 99.21%.
3. Berdasarkan uji coba yang dilakukan, panjang *frame* dan *overlapping frame* yang paling baik dalam merepresentasikan aktivitas pada dataset CCTV di Departemen Informatika ITS lantai 3 adalah panjang *frame L* = 15 dan *overlapping frame P* = 1 dengan nilai *accuracy* sebesar 95.80% dan rata-rata *recall* sebesar 95.42%.
4. Berdasarkan uji coba yang telah dilakukan, waktu pengambilan data dalam merepresentasikan aktivitas pada dataset CCTV di Departemen Informatika ITS lantai 3 untuk mencapai akurasi tertinggi adalah saat malam hari dimana

intensitas cahaya pada saat tersebut cenderung stabil oleh penerangan lampu.

6.2 Saran

Saran yang dapat diberikan dalam pengembangan lebih lanjut untuk menyempurnakan sistem adalah sebagai berikut.

Sebaiknya sistem pengenalan aktivitas pada video CCTV dapat dilakukan secara *realtime* sehingga dapat dimanfaatkan dalam bidang keamanan di Departemen Informatika ITS ini. Selain itu, sebaiknya pengembangan sistem dilakukan dengan memperbanyak kelas aktivitas yang dapat diklasifikasi seperti aktivitas dinamis lainnya, aktivitas statis seperti duduk dan berdiri, maupun kombinasi dari aktivitas-aktivitas tersebut, serta memungkinkan adanya pergerakan oleh lebih dari satu objek manusia.

DAFTAR PUSTAKA

[1] K. Guo, P. Ishwar and J. Konrad, "Action Recognition from Video Using Feature Covariance Matrices," in *Transaction on Image Processing, Vol. 22, No. 6*, IEEE, 2013, pp. 2479-2494.

[2] M. Ahmad, I. Parvin and S. W. Lee, "Silhouette history and energy image information for human movement recognition," in *J. Multimedia, vol. 5, no. 1*, 2010, pp. 12-21.

[3] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," in *IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 3*, 2001, pp. 257-267.

[4] Y. Chen, Q. Wu and X. He, "Human action recognition by Radon transform," in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2008, pp. 862-868.

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani and R. Basri, "Actions as space-time shapes," in *IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 12*, 2007, pp. 2247-2253.

[6] K. Guo, P. Ishwar and J. Konrad, "Action recognition from video by covariance matching of silhouette tunnels," in *Proc. Brazilian Symp. Comput. Graph. Image*, 2009, pp. 299-306.

[7] N. Ikizler and P. Duygulu, "Human action recognition using distribution of oriented rectangular patches," in *Proc. Human Motion, Understand. Model. Capture Animation*, 2007, pp. 271-284.

[8] L. Wang, H. Ning, T. Tan and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition," in *IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 2*, 2004, pp. 149-158.

[9] S. F. Wong and R. Cipolla, "Extracting spatio-temporal interest points using global information," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1-8.

[10] J. Yamato, J. Ohya and K. Ishii, "Recognizing human action in time sequential image using hidden markov model," in *IEEE Conf. Comput. Vis. Pattern Recognit*, 1992, pp. 379-385.

[11] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," in *IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 2*, 2010, pp. 288-303.

[12] S. Danafar and N. Gheissari, "Action recognition for surveillance applications using optic flow and SVM," in *Proc. Asian Conf. Comput. Vis.*, 2007, pp. 457-466.

[13] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1-8.

LAMPIRAN

L.1 *Confussion Matrix* Uji Coba Panjang *Frame L* = 8 pada Dataset Weizmann

		Nilai Prediksi									
		Run	***Walk***	***Wave2***	***Jump***	***Pjump***	***Jack***	***Side***	***Skip***	***Wave1***	***Bend***
Nilai Sebenarnya	*Run*	57	3	0	3	0	0	9	12	0	0
	Walk	3	130	0	7	0	1	5	15	0	0
	Wave2	0	0	106	0	5	10	0	0	17	2
	Jump	5	18	0	50	0	0	10	15	0	0
	Pjump	0	0	11	0	87	13	0	0	7	2
	Jack	0	0	5	0	2	158	0	0	0	0
	Side	14	14	0	12	0	1	41	14	0	0
	Skip	15	7	0	17	0	0	16	48	0	0
	Wave1	0	0	28	0	4	0	0	0	112	4
	Bend	0	0	2	0	4	0	0	1	2	135

L.2 *Confussion Matrix* Uji Coba Panjang *Frame L* = 20 pada Dataset Weizmann

		Nilai Prediksi									
		Run	***Walk***	***Wave2***	***Jump***	***Pjump***	***Jack***	***Side***	***Skip***	***Wave1***	***Bend***
Nilai Sebenarnya	*Run*	51	0	0	1	0	0	1	1	0	0
	Walk	0	130	0	0	0	0	0	1	0	0
	Wave2	0	0	110	0	2	0	0	0	1	0
	Jump	0	7	0	60	0	0	1	3	0	0
	Pjump	0	0	0	0	93	0	0	0	0	0
	Jack	0	0	0	0	0	138	0	0	0	0
	Side	3	6	0	0	0	0	57	3	0	0
	Skip	3	0	0	1	0	0	2	67	0	0
	Wave1	0	0	4	0	0	0	0	0	117	0
	Bend	0	0	2	0	0	0	0	0	2	113

L.3 *Confussion Matrix* Uji Coba *Overlapping Frame P* = 1 pada Dataset Weizmann

		Nilai Prediksi									
		Run	***Walk***	***Wave2***	***Jump***	***Pjump***	***Jack***	***Side***	***Skip***	***Wave1***	***Bend***
Nilai Sebenarnya	*Run*	213	0	0	0	0	0	1	1	0	0
	Walk	0	518	0	1	0	0	1	0	0	0
	Wave2	0	0	451	0	0	1	0	0	1	0
	Jump	1	4	0	281	0	0	0	1	0	0
	Pjump	0	0	0	0	367	0	0	0	0	0
	Jack	0	0	1	0	0	557	0	0	0	0
	Side	0	2	0	0	0	0	267	4	0	0
	Skip	1	2	0	1	0	0	1	291	0	0
	Wave1	0	0	2	0	0	0	0	0	479	1
	Bend	0	0	2	0	0	0	0	0	2	464

L.4 *Confussion Matrix* Uji Coba *Overlapping Frame P* = 2 pada Dataset Weizmann

		Nilai Prediksi									
		Run	***Walk***	***Wave2***	***Jump***	***Pjump***	***Jack***	***Side***	***Skip***	***Wave1***	***Bend***
Nilai Sebenarnya	*Run*	107	0	0	0	0	0	1	0	0	0
	Walk	0	255	0	3	0	0	1	1	0	0
	Wave2	0	0	225	0	0	1	0	0	0	0
	Jump	1	2	0	138	0	0	2	1	0	0
	Pjump	0	0	0	0	184	0	0	0	0	0
	Jack	0	0	0	0	0	278	0	0	0	0
	Side	1	4	0	1	0	0	128	3	0	0
	Skip	3	0	0	1	0	0	1	142	0	0
	Wave1	0	0	3	0	0	0	0	0	238	0
	Bend	0	0	1	0	0	0	0	0	4	229

L.5 *Confussion Matrix* Uji Coba *Overlapping Frame P* = 4 pada Dataset Weizmann

		Nilai Prediksi									
		Run	***Walk***	***Wave2***	***Jump***	***Pjump***	***Jack***	***Side***	***Skip***	***Wave1***	***Bend***
Nilai Sebenarnya	*Run*	51	0	0	1	0	0	1	1	0	0
	Walk	0	130	0	0	0	0	0	1	0	0
	Wave2	0	0	110	0	2	0	0	0	1	0
	Jump	0	7	0	60	0	0	1	3	0	0
	Pjump	0	0	0	0	93	0	0	0	0	0
	Jack	0	0	0	0	0	138	0	0	0	0
	Side	3	6	0	0	0	0	57	3	0	0
	Skip	3	0	0	1	0	0	2	67	0	0
	Wave1	0	0	4	0	0	0	0	0	117	0
	Bend	0	0	2	0	0	0	0	0	2	113

L.6 *Confussion Matrix* Uji Coba *Overlapping Frame P* = 6 pada Dataset Weizmann

		Nilai Prediksi									
		Run	***Walk***	***Wave2***	***Jump***	***Pjump***	***Jack***	***Side***	***Skip***	***Wave1***	***Bend***
Nilai Sebenarnya	*Run*	34	0	0	0	0	0	2	1	0	0
	Walk	0	83	0	0	0	0	1	2	0	0
	Wave2	0	0	71	0	2	1	0	0	1	0
	Jump	1	4	0	38	0	0	1	4	0	0
	Pjump	0	0	0	0	59	2	0	0	0	0
	Jack	0	0	1	0	0	92	0	0	0	0
	Side	1	3	0	0	0	0	39	3	0	0
	Skip	2	0	0	3	0	0	4	40	0	0
	Wave1	0	0	2	0	0	0	0	0	77	1
	Bend	0	0	0	0	2	0	0	0	2	74

L.7 *Confussion Matrix* Uji Coba Panjang *Frame L* = 10 pada Dataset CCTV

		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	113	136	40
	Berjalan	126	333	68
	Melambai	15	30	336

L.8 *Confussion Matrix* Uji Coba Panjang *Frame L* = 15 pada Dataset CCTV

		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	119	114	11
	Berjalan	78	388	16
	Melambai	3	3	345

L.9 *Confussion Matrix* Uji Coba Panjang *Frame L* = 20 pada Dataset CCTV

		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	80	101	18
	Berjalan	79	325	33
	Melambai	6	8	307

L.10 *Confussion Matrix* Uji Coba *Overlapping Frame P* = 1 pada Dataset CCTV

		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	890	76	7
	Berjalan	83	1837	9
	Melambai	4	2	1399

L.11 *Confussion Matrix* Uji Coba *Overlapping Frame P* = 2 pada Dataset CCTV

		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	405	73	9
	Berjalan	72	882	10
	Melambai	4	2	697

L.12 *Confussion Matrix* Uji Coba *Overlapping Frame P* = 4 pada Dataset CCTV

		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	119	114	11
	Berjalan	78	388	16
	Melambai	3	3	345

L.13 *Confussion Matrix* Uji Coba *Overlapping Frame P* = 6 pada Dataset CCTV

		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	77	77	9
	Berjalan	60	242	19
	Melambai	3	2	229

L.14 *Confussion Matrix* Uji Coba Waktu Pengambilan Data saat Pagi Hari pada Dataset CCTV

		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	1043	164	40
	Berjalan	130	2086	41
	Melambai	20	14	1571

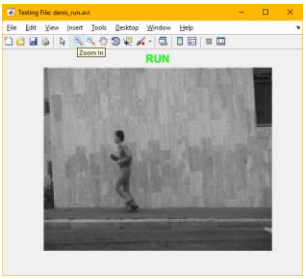
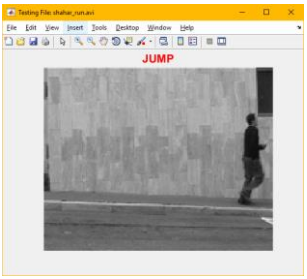
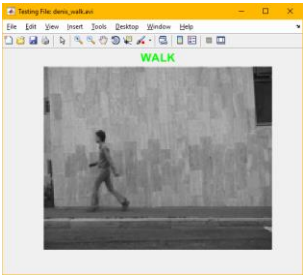
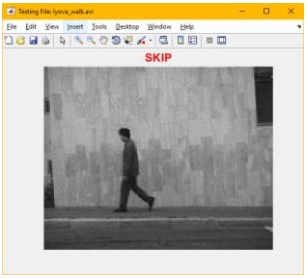
L.15 *Confussion Matrix* Uji Coba Waktu Pengambilan Data saat Siang Hari pada Dataset CCTV

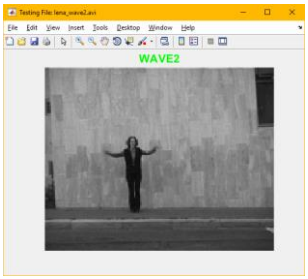
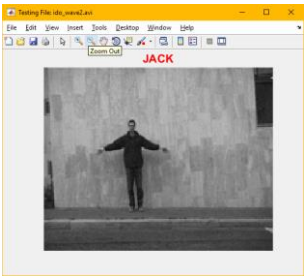
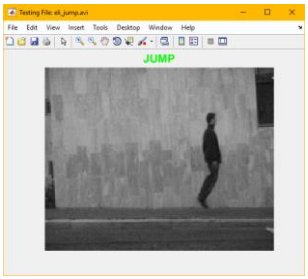
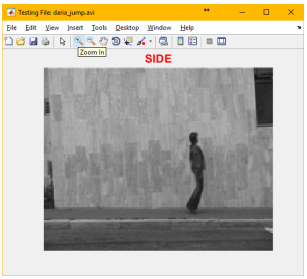
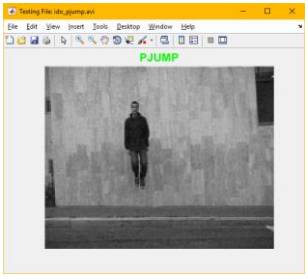
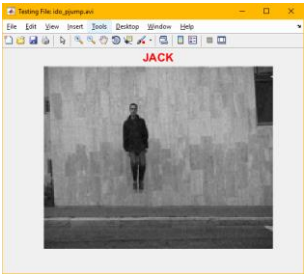
		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	890	76	7
	Berjalan	83	1837	9
	Melambai	4	2	1399

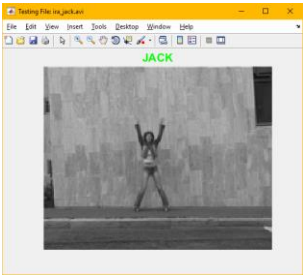
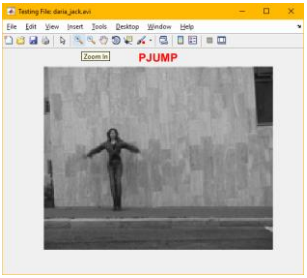
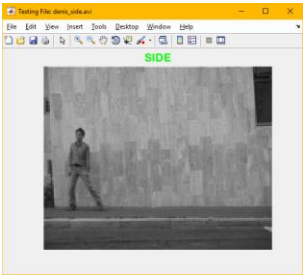
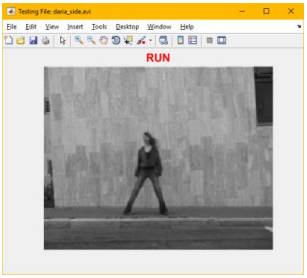
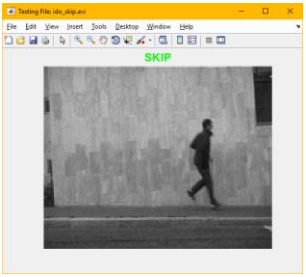
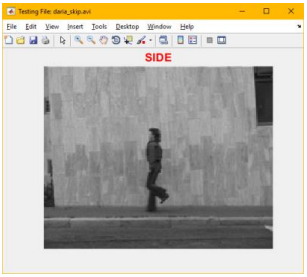
L.16 *Confussion Matrix* Uji Coba Waktu Pengambilan Data saat Malam Hari pada Dataset CCTV

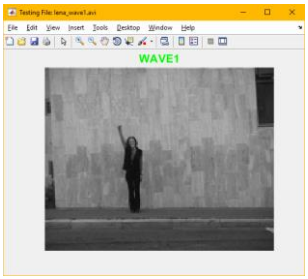
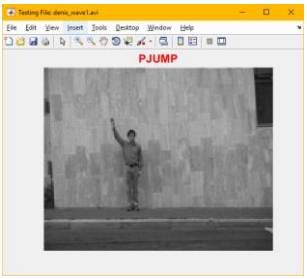
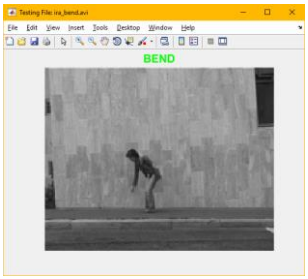
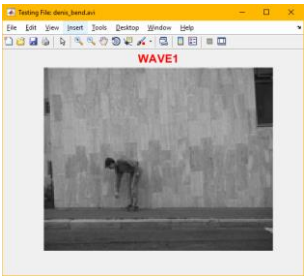
		Nilai Prediksi		
		Berlari	**Berjalan**	**Melambai**
Nilai Sebenarnya	**Berlari**	2647	157	25
	Berjalan	142	4169	14
	Melambai	9	6	1356

L.17 Gambar Hasil Klasifikasi Uji Coba pada Dataset Weizmann

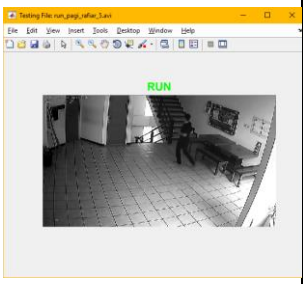
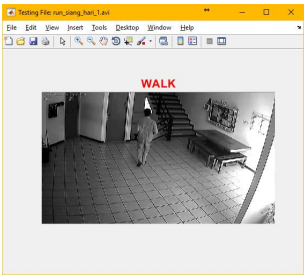
Kelas	Klasifikasi Benar	Klasifikasi Salah
Run		
Walk		

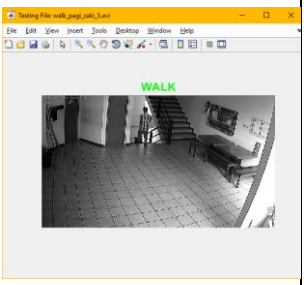
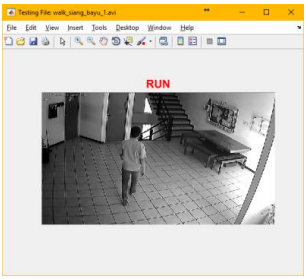
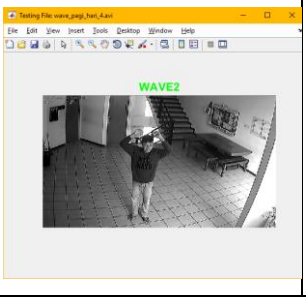
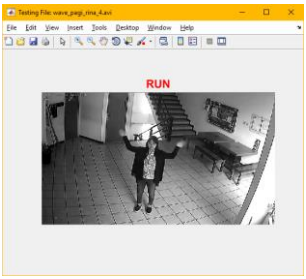
Kelas	Klasifikasi Benar	Klasifikasi Salah
Wave2		
Jump		
Pjump		

Kelas	Klasifikasi Benar	Klasifikasi Salah
Jack		
Side		
Skip		

Kelas	Klasifikasi Benar	Klasifikasi Salah
Wave1		
Bend		

L.18 Gambar Hasil Klasifikasi Uji Coba pada Dataset CCTV

Kelas	Klasifikasi Benar	Klasifikasi Salah
Berlari		

Kelas	Klasifikasi Benar	Klasifikasi Salah
Berjalan		
Melambai		

BIODATA PENULIS

Penulis, **Rina Wijaya Kusuma Wardhani**, lahir di Denpasar, 14 Mei 1996. Penulis menempuh pendidikan sekolah dasar di SD 19 Pemecutan Kota Denpasar. Melanjutkan pendidikan sekolah menengah pertama di SMP Negeri 7 Kota Denpasar dan selanjutnya di SMA Negeri 4 Kota Denpasar. Selanjutnya penulis melanjutkan pendidikan sarjana di Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya.

Selama kuliah, penulis aktif menjadi administrator Laboratorium Pemrograman, asisten mata kuliah Matematika Diskrit, asisten praktikum Sistem Basis Data, asisten mata kuliah Animasi dan Pemodelan 3D serta aktif dalam mengikuti beberapa lomba di bidang IT, organisasi tingkat jurusan, dan kepanitiaan. Diantaranya penulis berpartisipasi sebagai anggota Departemen Riset dan Teknologi HMTC-Informatika (2015-2016), anggota Departemen Teknologi HMTC-Informatika (2016-2017), anggota biro Web dan Kesekretariatan Schematics 2015 dan Wakil Koordinator biro Web dan Kesekretariatan Schematics 2016, dan anggota Klinik Mural ITS Expo 2015.

Penulis juga mengikuti kegiatan pelatihan, diantaranya berpartisipasi sebagai peserta aktif LKMM Pra Tingkat Dasar FTIf 2014, peserta aktif LKMM Tingkat Dasar FTIF 2015, dan peserta aktif Sekolah Himpunan HTMC-Informatika 2016.

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Komputasi Cerdas dan Visi (KCV). Penulis dapat dihubungi melalui email: rinawkw@gmail.com